RESEARCH CENTER
Paris - Rocquencourt

FIELD

Activity Report 2012

Section Scientific Foundations

Edition: 2013-04-24

ABSTRACTION Project-Team

3. Scientific Foundations

3.1. Abstract Interpretation Theory

The abstract interpretation theory [41], [32], [42], is the main scientific foundation of the work of the ABSTRACTION project-team. Its main current application is on the safety and security of complex hardware and software computer systems either sequential [41], [34], or parallel [36] with shared memory [33], [35], [44] or synchronous message [43] communication.

Abstract interpretation is a theory of sound approximation of mathematical structures, in particular those involved in the behavior of computer systems. It allows the systematic derivation of sound methods and algorithms for approximating undecidable or highly complex problems in various areas of computer science (semantics, verification and proof, model-checking, static analysis, program transformation and optimization, typing, software steganography, etc...) and system biology (pathways analysis).

3.2. Formal Verification by Abstract Interpretation

The *formal verification* of a program (and more generally a computer system) consists in proving that its *semantics* (describing "what the program executions actually do") satisfies its *specification* (describing "what the program executions are supposed to do").

Abstract interpretation formalizes the idea that this formal proof can be done at some level of abstraction where irrelevant details about the semantics and the specification are ignored. This amounts to proving that an *abstract semantics* satisfies an *abstract specification*. An example of abstract semantics is Hoare logic while examples of abstract specifications are invariance, partial, or total correctness. These examples abstract away from concrete properties such as execution times.

Abstractions should preferably be *sound* (no conclusion derived from the abstract semantics is wrong with respect to the program concrete semantics and specification). Otherwise stated, a proof that the abstract semantics satisfies the abstract specification should imply that the concrete semantics also satisfies the concrete specification. Hoare logic is a sound verification method, debugging is not (since some executions are left out), bounded model checking is not either (since parts of some executions are left out). Unsound abstractions lead to *false negatives* (the program may be claimed to be correct/non erroneous with respect to the specification whereas it is in fact incorrect). Abstract interpretation can be used to design sound semantics and formal verification methods (thus eliminating all false negatives).

Abstractions should also preferably be *complete* (no aspect of the semantics relevant to the specification is left out). So if the concrete semantics satisfies the concrete specification this should be provable in the abstract. However program proofs (for non-trivial program properties such as safety, liveness, or security) are undecidable. Nevertheless, we can design tools that address undecidable problems by allowing the tool not to terminate, to be driven by human intervention, to be unsound (e.g. debugging tools omit possible executions), or to be incomplete (e.g. static analysis tools may produce false alarms). Incomplete abstractions lead to *false positives* or *false alarms* (the specification is claimed to be potentially violated by some program executions while it is not). Semantics and formal verification methods designed by abstract interpretation may be complete (e.g. [39], [40], [47]) or incomplete (e.g. [2]).

Sound, automatic, terminating and precise tools are difficult to design. Complete automatic tools to solve non-trivial verification problems cannot exist, by undecidability. However static analysis tools producing very few or no false alarms have been designed and used in industrial contexts for specific families of properties and programs [45]. In all cases, abstract interpretation provides a systematic construction method based on the effective approximation of the concrete semantics, which can be (partly) automated and/or formally verified.

Abstract interpretation aims at:

- providing a basic coherent and conceptual theory for understanding in a unified framework the multiplicity of ideas, concepts, reasonings, methods, and tools on formal program analysis and verification [41], [42];
- guiding the correct formal design of *abstract semantics* [40], [47] and automatic tools for *program analysis* (computing an abstract semantics) and *program verification* (proving that an abstract semantics satisfies an abstract specification) [37].

Abstract interpretation theory studies semantics (formal models of computer systems), abstractions, their soundness, and completeness.

In practice, abstract interpretation is used to design analysis, compilation, optimization, and verification tools which must automatically and statically determine properties about the runtime behavior of programs. For example the ASTRÉE static analyzer (Section 5.2), which was developed by the team over the last decade, aims at proving the absence of runtime errors in programs written in the C programming language. It was originally used in the aerospace industry to verify very large, synchronous, time-triggered, real-time, safety-critical, embedded software and its scope of application was later broadly widened. ASTRÉE is now industrialized by AbsInt Angewandte Informatik GmbH and is commercially available.

3.3. Advanced Introductions to Abstract Interpretation

A short, informal, and intuitive introduction to the theory of abstract interpretation can be found in [37], see also "AbstractInterpretationinaNutshell" [1] on the web. A more comprehensive introduction is available online [2]. The paper entitled "Basicconceptsofabstractinterpretation" [38] and an elementary "courseonabstract interpretation" [3] can also be found on the web.

[1] www.di.ens.fr/~cousot/AI/IntroAbsInt.html
[2] www.di.ens.fr/~cousot/AI/
[3] web.mit.edu/afs/athena.mit.edu/course/16/16.399/www/

AOSTE Project-Team

3. Scientific Foundations

3.1. Models of Computation and Communication (MoCCs)

Participants: Charles André, Robert de Simone, Jean-Vivien Millo, Dumitru Potop Butucaru.

Esterel, SyncCharts, synchronous formalisms, Process Networks, Marked Graphs, Kahn networks, compilation, synthesis, formal verification, optimization, allocation, refinement, scheduling

Formal Models of Computation form the basis of our approach to Embedded System Design. Because of the growing importance of communication handling, it is now associated with the name, MoCC in short. The appeal of MoCCs comes from the fact that they combine features of mathematical models (formal analysis, transformation, and verification) with this of executable specifications (close to code level, simulation, and implementation). Examples of MoCCs in our case are mainly synchronous reactive formalisms and dataflow process networks. Various extensions or specific restrictions enforce respectively greater expressivity or more focused decidable analysis results.

DataFlow Process Networks and Synchronous Reactive Languages such as ESTEREL/SYNCCHARTS and SIGNAL/POLYCHRONY [53], [54], [48], [15], [4], [13] share one main characteristics: they are specified in a self-timed or loosely timed fashion, in the asynchronous data-flow style. But formal criteria in their semantics ensure that, under good correctness conditions, a sound synchronous interpretation can be provided, in which all treatments (computations, signaling communications) are precisely temporally mapped. This is refered to as clock calculus in synchronous reactive systems, and leads to a large body of theoretical studies and deep results in the case of DataFlow Process Networks [49], [47] (consider SDF balance equations for instance [56]).

As a result, explicit schedules become an important ingredient of design, which ultimately can be considered and handled by the designer him/herself. In practice such schedules are sought to optimize other parts of the design, mainly buffering queues: production and consumption of data can be regulated in their relative speeds. This was specially taken into account in the recent theories of Latency-Insensitive Design [50], or N-synchronous processes [51], with some of our contributions [6].

Explicit schedule patterns should be pictured in the framework of low-power distributed mapping of embedded applications onto manycore architectures, where they could play an important role as theoretical formal models on which to compute and optimize allocations and performances. We describe below two lines of research in this direction. Striking in these techniques is the fact that they include time and timing as integral parts of early functional design. But this original time is logical, multiform, and only partially ordering the various functional computations and communications. This approach was radically generalized in our team to a methodology for logical time based design, described next (see 3.2).

3.1.1. K-periodic static scheduling and routing in Process Networks

In the recent years we focused on the algorithm treatments of ultimately k-periodic schedule regimes, which are the class of schedules obtained by many of the theories described above. An important breakthrough occurred when realizing that the type of ultimatelly periodic binary words that were used for reporting *static scheduling* results could also be employed to record a completely distinct notion of ultimately k-periodic route switching patterns, and furthermore that commonalities of representation could ease combine them together. A new model, by the name of K-periodical Routed marked Graphs (KRG) was introduced, and extensively studied for algebraic and algorithmic properties [5].

The computations of optimized static schedules and other optimal buffering configurations in the context of latency-insensitive design led to the K-Passa software tool development 5.2 .

3.1.2. Endochrony and GALS implementation of conflict-free polychronous programs

The possibility of exploring various schedulings for a given application comes from the fact that some behaviors are truly concurrent, and mutually *conflict-free* (so they can be executed independently, with any choice of ordering). Discovering potential asynchronous inside synchronous reactive specifications then becomes something highly desirable. It can benefit to potential distributed implementation, where signal communications are restricted to a minimum, as they usually incur loss in performance and higher power consumption. This general line of research has come to be known as Endochrony, with some of our contributions [11].

3.2. Logical Time in Model-Driven Embedded System Design

Participants: Charles André, Julien deAntoni, Frédéric Mallet, Marie-Agnès Peraldi Frati, Robert de Simone.

Starting from specific needs and opportunities for formal design of embedded systems as learned from our work on MoCCs (see 3.1), we developed a Logical Time Model as part of the official OMG UML profile MARTE for Modeling and Analysis of Real-Time Embedded systems. With this model is associated a Clock Constraint Specification Language (CCSL), which allows to provide loose or strict logical time constraints between design ingredients, be them computations, communications, or any kind of events whose repetitions can be conceived as generating a logical conceptual clock (or activation condition). The definition of CCSL is provided in [1].

Our vision is that many (if not all) of the timing constraints generally expressed as physical prescriptions in real-time embedded design (such as periodicity, sporadicity) could be expressed in a logical setting, while actually many physical timing values are still unknown or unspecified at this stage. On the other hand, our logical view may express much more, such as loosely stated timing relations based on partial orderings or partial constraints.

So far we have used CCSL to express important phenonema as present in several formalisms: AADL (used in avionics domain), EAST-ADL2 (proposed for the AutoSar automotive electronic design approach), IP-Xact (for System-on-Chip (*SoC*) design). The difference here comes from the fact that these formalisms were formerly describing such issues in informal terms, while CCSL provides a dedicated formal mathematical notation. Close connections with synchronous and polychronous languages, especially Signal, were also established; so was the ability of CCSL to model dataflow process network static scheduling.

In principle the MARTE profile and its Logical Time Model can be used with any UML editor supporting profiles. In practice we focused on the PAPYRUS open-source editor, mainly from CEA LIST. We developed under Eclipse the TIMESQUARE solver and emulator for CCSL constraints (see 5.1), with its own graphical interface, as a stand-alone software module, while strongly coupled with MARTE and Papyrus.

While CCSL constraints may be introduced as part of the intended functionality, some may also be extracted from requirements imposed either from real-time user demands, or from the resource limitations and features from the intended execution platform. Sophisticated detailed descriptions of platform architectures are allowed using MARTE, as well as formal allocations of application operations (computations and communications) onto platform resources (processors and interconnects). This is of course of great value at a time where embedded architectures are becoming more and more heterogeneous and parallel or distributed, so that application mapping in terms of spatial allocation and temporal scheduling becomes harder and harder. This approach is extensively supported by the MARTE profile and its various models. As such it originates from the Application-Architecture-Adequation (AAA) methodology, first proposed by Yves Sorel, member of Aoste. AAA aims at specific distributed real-time algorithmic methods, described next in 3.3 .

Of course, while logical time in design is promoted here, and our works show how many current notions used in real-time and embedded systems synthesis can naturally be phrased in this model, there will be in the end a phase of validation of the logical time assumptions (as is the case in synchronous circuits and SoC design with timing closure issues). This validation is usually conducted from Worst-Case Execution Time (WCET) analysis on individual components, which are then used in further analysis techniques to establish the validity of logical time assumptions (as partial constraints) asserted during the design.

3.3. The AAA (Algorithm-Architecture Adequation) methodology and Real-Time Scheduling

Participants: Laurent George, Dumitru Potop Butucaru, Yves Sorel.

Note: The AAA methodology and the SynDEx environment are fully described at http://www.syndex.org/, together with relevant publications.

3.3.1. Algorithm-Architecture Adequation

The AAA methodology relies on distributed real-time scheduling and relevant optimization to connect an Algorithm/Application model to an Architectural one. We now describe its premises and benefits.

The Algorithm model is an extension of the well known data-flow model from Dennis [52]. It is a directed acyclic hyper-graph (DAG) that we call "conditioned factorized data dependence graph", whose vertices are "operations" and hyper-edges are directed "data or control dependences" between operations. The data dependences define a partial order on the operations execution. The basic data-flow model was extended in three directions: first infinite (resp. finite) repetition of a sub-graph pattern in order to specify the reactive aspect of real-time systems (resp. in order to specify the finite repetition of a sub-graph consuming different data similar to a loop in imperative languages), second "state" when data dependences are necessary between different infinite repetitions of the sub-graph pattern introducing cycles which must be avoided by introducing specific vertices called "delays" (similar to z^{-n} in automatic control), third "conditioning" of an operation by a control dependence similar to conditional control structure in imperative languages, allowing the execution of alternative subgraphs. Delays combined with conditioning allow the programmer to specify automata necessary for describing "mode changes".

The Architecture model is a directed graph, whose vertices are of two types: "processor" (one sequencer of operations and possibly several sequencers of communications) and "medium" (support of communications), and whose edges are directed connections.

The resulting implementation model [9] is obtained by an external compositional law, for which the architecture graph operates on the algorithm graph. Thus, that result is a set of algorithm graphs, "architecture-aware", corresponding to refinements of the initial algorithm graph, by computing spatial (distribution) and timing (scheduling) allocations of the operations according to the architecture graph resource availability. In that context "Adequation" refers to some search amongst the solution space of resulting algorithm graphs, labelled by timing characteristics, for one which verifies timing constraints and optimizes some criteria, usually the total execution time and the number of computing resources (but other criteria may exist). The next section describes distributed real-time schedulability analysis and optimization techniques for that purpose.

3.3.2. Distributed Real-Time Scheduling and Optimization

We address two main issues: monoprocessor real-time scheduling and multiprocessor real-time scheduling where constraints must mandatorily be met otherwise dramatic consequences may occur (hard real-time) and where resources must be minimized because of embedded features.

In our monoprocessor real-time scheduling work, beside the classical deadline constraint, often equal to a period, we take into consideration dependences beetween tasks and several, possibly related, latencies. A latency is a generalization of the typical "end-to-end" constraint. Dealing with multiple real-time constraints raises the complexity of that issue. Moreover, because the preemption leads to a waste of resources due to its approximation in the WCET (Worst Execution Time) of every task as proposed by Liu and Leyland [57], we first studied non-preemtive real-time scheduling with dependences, periodicities, and latencies constraints. Although a bad approximation may have dramatic consequences on real-time scheduling, there are only few researches on this topic. We have been investigating preemptive real-time scheduling since few years, but seeking the exact cost of the preemption such that it can be integrated in schedulability conditions, and in the corresponding scheduling algorithms. More generally, we are interested in integrating in the schedulability analyses the cost of the RTOS (Real-Time Operating System), for which the exact cost of preemption is the most difficult part because it varies according to the instance of each task [10]. Finally, we investigate also the problem of mixing hard real-time and soft real-time constraints that arises in the most complex applications.

The second research area is devoted to distributed real-time scheduling with embedding constraints. We use the results obtained in the monoprocessor case in order to derive solutions for the problem of multiprocessor (distributed) real-time scheduling. In addition to satisfy the multiple real-time constraints mentioned in the monoprocessor case, we have to minimize the total execution time (makespan) since we deal with automatic control applications involving feedback. Furthermore, the domain of embedded systems leads to solving minimization resources problems. Since these optimization problems are of NP-hard complexity we develop exact algorithms (B & B, B & C) which are optimal for simple problems, and heuristics which are sub-optimal for realistic problems corresponding to industrial needs. Long time ago we proposed a very fast "greedy" heuristics [8] whose results were regularly improved, and extended with local neighborhood heuristics, or used as initial solutions for metaheuristics such as variants of "simulated annealing".

In addition to the spatial dimension (distributed) of the real-time scheduling problem, other important dimensions are the type of communication mechanisms (shared memory vs. message passing), or the source of control and synchronization (event-driven vs. time-triggered). We explore real-time scheduling on architectures corresponding to all combinations of the above dimensions. This is of particular impact in application domains such as automotive and avionics (see 4.2).

Since real-time distributed systems are often safety-critical we address dependability issues, to tolerate faults in processors and communication interconnects. We maily focus on software redondancy, rather than hardware, to ensure real-time behaviour preservation in presence of faulty processors and/or communication media (where possible failures are predictively specified by the designer). We investigate fail silent, transient, intermittent, and Byzantine faults.

CASCADE Project-Team

3. Scientific Foundations

3.1. Provable Security

Since the beginning of public-key cryptography, with the seminal Diffie-Hellman paper [75], many suitable algorithmic problems for cryptography have been proposed and many cryptographic schemes have been designed, together with more or less heuristic proofs of their security relative to the intractability of the underlying problems. However, many of those schemes have thereafter been broken. The simple fact that a cryptographic algorithm withstood cryptanalytic attacks for several years has often been considered as a kind of validation procedure, but schemes may take a long time before being broken. An example is the Chor-Rivest cryptosystem [74], based on the knapsack problem, which took more than 10 years to be totally broken [90], whereas before this attack it was believed to be strongly secure. As a consequence, the lack of attacks at some time should never be considered as a full security validation of the proposal.

A completely different paradigm is provided by the concept of "provable" security. A significant line of research has tried to provide proofs in the framework of complexity theory (a.k.a. "reductionist" security proofs): the proofs provide reductions from a well-studied problem (factoring, RSA or the discrete logarithm) to an attack against a cryptographic protocol.

At the beginning, researchers just tried to define the security notions required by actual cryptographic schemes, and then to design protocols which could achieve these notions. The techniques were directly derived from complexity theory, providing polynomial reductions. However, their aim was essentially theoretical. They were indeed trying to minimize the required assumptions on the primitives (one-way functions or permutations, possibly trapdoor, etc) [78], without considering practicality. Therefore, they just needed to design a scheme with polynomial-time algorithms, and to exhibit polynomial reductions from the basic mathematical assumption on the hardness of the underlying problem into an attack of the security notion, in an asymptotic way. However, such a result has no practical impact on actual security. Indeed, even with a polynomial reduction, one may be able to break the cryptographic protocol within a few hours, whereas the reduction just leads to an algorithm against the underlying problem which requires many years. Therefore, those reductions only prove the security when very huge (and thus maybe unpractical) parameters are in use, under the assumption that no polynomial time algorithm exists to solve the underlying problem.

For a few years, more efficient reductions have been expected, under the denomination of either "exact security" [71] or "concrete security" [83], which provide more practical security results. The perfect situation is reached when one is able to prove that, from an attack, one can describe an algorithm against the underlying problem, with almost the same success probability within almost the same amount of time: "tight reductions". We have then achieved "practical security" [67].

Unfortunately, in many cases, even just provable security is at the cost of an important loss in terms of efficiency for the cryptographic protocol. Thus, some models have been proposed, trying to deal with the security of efficient schemes: some concrete objects are identified with ideal (or black-box) ones. For example, it is by now usual to identify hash functions with ideal random functions, in the so-called "random-oracle model", informally introduced by Fiat and Shamir [76], and later formalized by Bellare and Rogaway [70]. Similarly, block ciphers are identified with families of truly random permutations in the "ideal cipher model" [68]. A few years ago, another kind of idealization was introduced in cryptography, the black-box group, where the group operation, in any algebraic group, is defined by a black-box: a new element necessarily comes from the addition (or the subtraction) of two already known elements. It is by now called the "generic model" [82], [89]. Some works even require several ideal models together to provide some new validations [73].

More recently, the new trend is to get provable security, without such ideal assumptions (there are currently a long list of publications showing "without random oracles" in their title), but under new and possibly stronger computational assumptions. As a consequence, a cryptographer has to deal with the three following important steps:

computational assumptions, which are the foundations of the security. We thus need to have a strong evidence that the computational problems are reasonably hard to solve. We study several assumptions, by improving algorithms (attacks), and notably using lattice reductions. We furthermore contribute to the list of "potential" hard problems.

security model, which makes precise the security notions one wants to achieve, as well as the means the adversary may be given. We contribute to this point, in several ways:

- by providing a security model for many primitives and protocols, and namely group-oriented protocols, which involve many parties, but also many communications (group key exchange, group signatures, etc);
- by enhancing some classical security models;
- by considering new means for the adversary, such as side-channel information.

design of new schemes/protocols, or more efficient, with additional features, etc.

security proof, which consists in exhibiting a reduction.

For a long time, the security proofs by reduction used classical techniques from complexity theory, with a direct description of the reduction, and then a long and quite technical analysis for providing the probabilistic estimates. Such analysis is unfortunately error-prone. Victor Shoup proposed a nice way to organize the proofs, and eventually obtain the probabilities, using a sequence of games [88], [69], [84] which highlights the computational assumptions, and splits the analysis in small independent problems. We early adopted and developed this technique, and namely in [77]. We applied this methodology to various kinds of systems, in order to achieve the highest security properties: authenticity, integrity, confidentiality, privacy, anonymity. Nevertheless, efficiency was also a basic requirement.

However, such reductions are notoriously error-prone: errors have been found in many published protocols. Security errors can have serious consequences, such as loss of money in the case of electronic commerce. Moreover, security errors cannot be detected by testing, because they appear only in the presence of a malicious adversary.

Security protocols are therefore an important area for formal verification.

3.2. Cryptanalysis

Because there is no absolute proof of security, it is essential to study cryptanalysis, which is roughly speaking the science of code-breaking. As a result, key-sizes are usually selected based on the state-of-the-art in cryptanalysis. The previous section emphasized that public-key cryptography required hard computational problems: if there is no hard problem, there cannot be any public-key cryptography either. If any of the computational problems mentioned above turns out to be easy to solve, then the corresponding cryptosystems can be broken, as the public key would actually disclose the private key. This means that one obvious way to cryptanalyze is to solve the underlying algorithmic problems, such as integer factorization, discrete logarithm, lattice reduction, Gröbner bases, *etc.* Here, we mean a study of the computational problem in its full generality. The project-team has a strong expertise (both in design and analysis) on the best algorithms for lattice reduction, which are also very useful to attack classical schemes based on factorization or discrete logarithm.

Alternatively, one may try to exploit the special properties of the cryptographic instances of the computational problem. Even if the underlying general problem is NP-hard, its cryptographic instances may be much easier, because the cryptographic functionalities typically require a specific mathematical structure. In particular, this means that there might be an attack which can only be used to break the scheme, but not to solve the underlying problem in general. This happened many times in knapsack cryptography and multivariate cryptography. Interestingly, generic tools to solve the general problem perform sometimes even much better on cryptographic instances (this happened for Gröbner bases and lattice reduction).

However, if the underlying computational problem turns out to be really hard both in general and for instances of cryptographic interest, this will not necessarily imply that the cryptosystem is secure. First of all, it is not even clear what is meant exactly by the term *secure* or *insecure*. Should an encryption scheme which leaks the first bit of the plaintext be considered secure? Is the secret key really necessary to decrypt ciphertexts or to sign messages? If a cryptosystem is theoretically secure, could there be potential security flaws for its implementation? For instance, if some of the temporary variables (such as pseudo-random numbers) used during the cryptographic operations are partially leaked, could it have an impact on the security of the cryptosystem? This means that there is much more into cryptanalysis than just trying to solve the main algorithmic problems. In particular, cryptanalysts are interested in defining and studying realistic environments for attacks (adaptive chosen-ciphertext attacks, side-channel attacks, *etc.*), as well as goals of attacks (key recovery, partial information, existential forgery, distinguishability, *etc.*). As such, there are obvious connections with provable security. It is perhaps worth noting that cryptanalysis also proved to be a good incentive for the introduction of new techniques in cryptology. Indeed, several mathematical objects now considered invaluable in cryptographic design were first introduced in cryptology as cryptanalytic tools, including lattices and pairings. The project-team has a strong expertise in cryptanalysis: many schemes have been broken, and new techniques have been developed.

3.3. Symmetric Cryptography

Even if asymmetric cryptography has been a major breakthrough in cryptography, and a key element in its recent development, conventional cryptography (a.k.a. symmetric, or secret key cryptography) is still required in any application: asymmetric cryptography is much more powerful and convenient, since it allows signatures, key exchange, etc. However, it is not well-suited for high-rate communication links, such as video or audio streaming. Therefore, block-ciphers remain a fundamental primitive. However, since the AES Competition (which started in January 1997, and eventually selected the Rijndael algorithm in October 2000), this domain has become less active, even though some researchers are still trying to develop new attacks. On the opposite, because of the lack of widely admitted stream ciphers (able to encrypt high-speed streams of data), ECRYPT (the European Network of Excellence in Cryptology) launched the eSTREAM project, which investigated research on this topic, at the international level: many teams proposed candidates that have been analyzed by the entire cryptographic community. Similarly, in the last few years, hash functions [86], [85], [80], [81], [79], which are an essential primitive in many protocols, received a lot of attention: they were initially used for improving efficiency in signature schemes, hence the requirement of collision-resistance. But afterwards, hash functions have been used for many purposes, such as key derivation, random generation, and random functions (random oracles [70]). Recently, a bunch of attacks [72], [91], [92], [93], [94], [96], [95] have shown several drastic weaknesses on all known hash functions. Knowing more (how weak they are) about them, but also building new hash functions are major challenges. For the latter goal, the first task is to formally define a security model for hash functions, since no realistic formal model exists at the moment: in a way, we expect too much from hash functions, and it is therefore impossible to design such "ideal" functions. Because of the high priority of this goal (the design of a new hash function), the NIST has launched an international competition, called SHA-3 (similar to the AES competition 10 years ago), in order to select and standardize a hash function. Keccak has been officially chosen on October 2nd, 2012.

One way to design new hash functions may be a new mode of operation, which would involve a block cipher, iterated in a specific manner. This is already used to build stream ciphers and message authentication codes (symmetric authentication). Under some assumptions on the block cipher, it might be possible to apply the above methodology of provable security in order to prove the validity of the new design, according to a specific security model.

CONTRAINTES Project-Team

3. Scientific Foundations

3.1. Rule-based Modeling Languages

Logic programming in a broad sense is a declarative programming paradigm based on mathematical logic with the following identifications:

$$program = logical\ formula,$$

$$execution = proof\ search,$$

In Constraint Satisfaction Problems (CSP), the logical formulae are conjunctions of constraints (i.e. relations on variables expressing partial information) and the satisfiability proofs are computed by constraint solving procedures.

In Constraint Logic Programming (CLP), the logical formulae are Horn clauses with constraints (i.e. one headed rules for the inductive definitions of relations on variables) and the satisfiability proofs combine constraint solving and clause resolution. Gnu-Prolog and its modular extension EMoP that we develop, belong to this family of languages. We use them for solving combinatorial problems and for implementing Biocham.

In Concurrent Constraint Programming (CCP), CLP resolution is extended with a synchronization mechanism based on constraint entailment. The variables play the role of transmissible dynamically created communication channels. An agent may add constraints to the store or read the store to decide whether a constraint guard is entailed by the current store. Sicstus-Prolog and SWI-Prolog belong the this family of languages. We use them for solving combinatorial optimization problems and defining new global constraints.

CCP execution can be identified to deduction in J.Y. Girard's Linear Logic by interpreting multisets of constraints and agents as tensor product conjunctions and guards and rules as linear implications [1]. The logical completeness of CCP in LL continues to hold when considering linear logic constraint systems, i.e. constraint systems where constraints can be consumed by implication. This extension, named Linear Logic Concurrent Constraint Programming (LLCC), allows for a non-monotonic evolution of the store of constraints and can encode multi-headed rules like the Constraint Handling Rules (CHR) language of T. Frühwirth.

All these rule-based languages, of increasing expressivity, involve some form of *multiset rewriting*. We have designed and continue developing the following modeling languages:

- Rules2CP, a rule-based modeling language for solving constraint optimization problems, developed for non-programmers,
- SiLCC, our experimental implementation of LLCC,
- the Biochemical Abstract Machine BIOCHAM, a rule-based modeling language dedicated to Systems Biology, in which biochemical reactions between multisets of reactants and products are expressed with multi-headed rules (somewhat similar to CHR rules) and augmented with *kinetic expressions* from which one can derive quantitative interpretations by Ordinary Differential Equations (ODE), Continuous-Time Markov Chains (CTMC) or Hybrid Automata.

3.2. Constraint Solving Techniques

Constraint propagation algorithms use constraints actively during search for filtering the domains of variables and reducing the search space. These domain reductions are the only way constraints communicate between each other. Our research involves different constraint domains, namely:

- booleans: binary decision diagrams and SAT solvers;

[1] F. Fages, P. Ruet, S. Soliman. *Linear concurrent constraint programming: operational and phase semantics*, in "Information and Control", 2001, vol. 165(1), pp.14-41.

- finite domains (bounded natural numbers): membership, arithmetic, reified [20], higher order and global constraints;
- reals: polyhedral libraries for linear constraints and interval methods;
- terms: subtyping constraints;
- graphs: subgraph epimorphism (SEPI) and isomorphism constraints; acyclicity constraint;
- Petri nets: P/T-invariants [5], siphons and traps [10];
- Kripke structures: temporal logic constraints (first-order Computation Tree Logic constraints over the reals).

We develop new constraints and domain filtering algorithms by using already existing constraint solving algorithms and implementations. For instance, we use the Parma Polyhedra Library PPL with its interface with Prolog for solving temporal logic constraints over the reals. Similarly, we use standard finite domain constraints for developing solvers for the new SEPI graph constraint.

3.3. Formal Methods for Systems Biology

At the end of the 90s, research in Bioinformatics evolved, passing from the analysis of the genomic sequence to the analysis of post-genomic interaction networks (expression of RNA and proteins, protein-protein interactions, transport, etc.). Systems biology is the name given to a pluridisciplinary research field involving biology, computer science, mathematics, physics, to illustrate this change of focus towards system-level understanding of high-level functions of living organisms from their biochemical bases at the molecular level.

Our group was among the first ones in 2002 to apply formal methods from computer science to systems biology in order to reason on large molecular interaction networks and get over complexity walls. The *logical paradigm for systems biology* that we develop can be summarized by the following identifications :

$$\textit{biological model} = \textit{rule-based transition system},$$
$$\textit{biological property} = \textit{temporal logic formula},$$
$$\textit{model validation} = \textit{model-checking},$$
$$\textit{model inference} = \textit{constraint solving}.$$

Rule-based dynamical models of biochemical reaction networks are composed of a reaction graph (bipartite graph with vertices for species and reactions) where the reaction vertices are given with kinetic expressions (mass action law, Michaelis-Menten, Hill, etc.). Most of our work consists in analysing the *interplay between the structure* (reaction graphs) *and the dynamics* (ODE, CTMC or hybrid interpretations derived from the kinetic expressions).

Besides this logical paradigm, we use the theory of abstract interpretation to relate the different interpretations of rule-based models and organize them in a hierarchy of semantics from the most concrete (CTMC stochastic semantics) to the most abstract (asynchronous Boolean transition system). This allows us to prove for instance that if a behavior is not possible in the Boolean semantics of the rules then it is not possible in the stochastic semantics for any kinetic expressions and parameter values. We also use the framework of abstract interpretation to formally relate rule-based reaction models to other knowledge representation formalisms such as, for instance, ontologies of protein functions, or influence graphs between molecular species. These formal methods are used to build models of biological processes, fit models to experimental data, make predictions, and design new biological experiments.

3.4. Tight Integration of In Silico and In Vivo Approaches

Bridging the gap between the complexity of biological systems and our capacity to model and predict systems behaviors is a central challenge in quantitative systems biology. We investigate using wet and dry experiments a few challenging biological questions that necessitate a tight integration between *in vivo* and *in silico* work. Key to the success of this line of research fundamentally guided by specific biological questions is the deployment of innovative modelling and analysis methods for the *in silico* studies.

Synthetic biology, or bioengineering, aims at designing and constructing *in vivo* biological systems that performs novel, useful tasks. This is achieved by reingeneering existing natural biological systems. While the construction of simple intracellular circuits has shown the feasibility of the approach, the design of larger, multicellular systems is a major open issue. In engineered tissues for example, the behavior results from the subtle interplay between intracellular processes (signal transduction, gene expression) and intercellular processes (contact inhibition, gradient of diffusible molecule). How should cells be genetically modified such that the desired behavior robustly emerges from cell interactions? In collaboration with Dirk Drasdo (EPI BANG), we develop *abstraction methods for multiscale systems* to make the design and optimization of such systems computationally tractable and investigate the mammalian tissue homeostasis problem from a bioengineering point of view. Then, in collaboration with the Weiss lab (MIT), we construct and test *in vitro* the proposed designs in actively-growing mammalian cells.

The rational design of synthetic systems relies however on a good quantitative understanding of the functioning of the various processes involved. To acquire that knowledge, one observes the cell reaction to a range of external perturbations. However, current experimental techniques do not allow precise perturbations of cellular processes over a long time period. To make progress on this problem, we develop an experimental platform for the *closed-loop control* of intracellular processes. In collaboration with the MSC lab (CNRS/Paris Diderot U), we develop models of the controlled cellular system, generate quantitative data for parameter identification, and develop real-time control approaches. The integration of all these elements results in an original platform combining hardware (microfluidic device and microscope) and software (cell tracking and model predictive control algorithms). More specifically, by setting up an external, *in silico* feedback loop, we investigate the strengths and time scales of natural feedback loops, responsible for cell adaptation to environmental fluctuations.

DEDUCTEAM Team (section vide)

FORMES Team

3. Scientific Foundations

3.1. Rewriting and Type theory

Coq [42] is one of the most popular proof assistant, in the academia and in the industry. Based on the Extended Calculus of Inductive Constructions, Coq has four kinds of basic entities: objects are used for computations (data, programs, proofs are objects); types express properties of objects; kinds categorize types by their logical structure. Coq's type checker can decide whether a given object satisfies a given type, and if a given type has a logical structure expressed by a given kind. Because it is possible to (uniformly) define inductive types such as lists, dependent types such as lists-of-length-n, parametric types such as lists-of-something, inductive properties such as $(even\ n)$ for some natural number n, etc, writing small specifications in Coq is an easy task. Writing proofs is a harder (non-automatable) task that must be done by the user with the help of tactics. We are interested in two challenges that one has to face with the development of formal proofs in Coq: the theoretical status of equality on the one hand, and the confidence one may have in Coq's proofs on the other hand. Our answer to the first challenge is CoqMTU, which isolates equality in a theory T, which must be first order, such as Presburger Arithmetic. Our answer to the second challenge is the (manual) certification of CoqMTU in Coq.

Rewriting is at the heart of proof systems such as the Extended Calculus of Constructions on which Coq is based, since mathematical proofs are made of reasonning steps, expressed by the typing rules of a given proof system, and computational steps, expressed by its rewrite rules. The certification of a proof system involves, in particular, proving three main properties of its rewrite rules: subject reduction (rewriting should preserve types), confluence (computations should be deterministic), and termination -computations must always terminate. The fact that falsity is not provable in a given proof system such as CoqMTU follows from the previous properties, while decidability of type-checking may require further work. These meta-theoretical proofs are indeed very complex, although at the same time very repetitive, depending on both the typing rules and the rewrite rules. A challenging research question here is to develop certification tools aiming at automating these proofs. Building such tools requires new results allowing to check subject-reduction, confluence and termination of higher-order calculi that are found in proof systems. Since subject-reduction is usually easy to check while consistency and decidability of type-checking follow, *in general*, from the others, confluence and termination are two very active research topics in this area. A last challenge to achieve these goals is the formalization itself of proof systems.

3.2. Verification

Model checking is an automatic formal verification technique [38]. In order to apply the technique, users have to formally specify desired properties on an abstract model of the system under verification. Model checkers will check whether the abstract model satisfies the given properties. If model checkers are able to prove or disprove the properties on the abstract model, they report the result and terminate. In practice, however, abstract models can be extremely complicated, model checkers may not conclude with reasonable computational resources.

Compositional reasoning is a way to ameliorate the complexity in abstract models [75]. Compositional reasoning tries to prove global properties on abstract models by establishing local properties on their components. If local properties on components are easier to verify, compositional reasoning can improve the capacity of model checking by local reasoning. Experiences however suggest that local reasoning may not suffice to establish global properties. It is rare that a global property can be established without considering their interactions. In assume-guarantee reasoning, model checkers try to verify local properties under a contextual assumption of each component. If contextual assumptions faithfully capture interactions among components, model checkers can conclude the verification of global properties.

Finding contextual assumptions however is difficult and may require clairvoyance. Interestingly, a fully automated technique for computing contextual assumptions was proposed in [41]. The automated technique formalizes the contextual assumption generation problem as a learning problem. If properties and abstract models are formalized as finite automata, then a contextual assumption is nothing but an unknown finite automaton that characterizes the environment. Applying a learning algorithm for finite automata, the automated technique will generate contextual assumptions for assume-guarantee reasoning. Experimental results show that the automated technique can outperform a monolithic and explicit verification algorithm.

The success of the learning-based assume-guarantee reasoning is however not satisfactory. Most verification tools are using implicit algorithms. In fact, implicit representations such as Binary Decision Diagrams can improve the capacity of model checking algorithms in several order of magnitudes. Early learning-based techniques, on the other hand, are based on the L^* learning algorithm using explicit representations. If a contextual assumption requires hundreds of states, the learning algorithm will take too much time to infer an assumption. Subsequently, early learning-based techniques cannot compete with monolithic implicit verification [40].

Recently, we propose assume-guarantee reasoning with implicit learning [37]. Our idea is to adopt an implicit representation used in the learning-based framework. Instead of enumerating states of contextual assumptions explicitly, our new technique computes transition relations as an implicit representation of contextual assumptions. Using a learning algorithm for Boolean functions, the new technique can easily compute contextual assumptions with thousands of states. Our preliminary experimental results show that the implicit learning technique can outperform interpolation-based monolithic implicit model checking in several parametrized test cases such as synchronous bus arbiters and the MSI cache coherence protocol.

Learning Boolean functions can also be applied to loop invariant inference [53], [54]. Suppose that a programmer annotates a loop with pre- and post-conditions. We would like to compute a loop invariant to verify that the annotated loop conforms to its specification. Finding loop invariants manually is very tedious. One makes a first guess and then iteratively refines the guess by examining the loop body. This process is in fact very similar to learning an unknown formula. Applying predicate abstraction and decision procedures, a learning algorithm for Boolean functions can infer loop invariants generated by a given set of atomic predicates. Preliminary experimental results show that the learning-based technique is effective for annotated loops extracted from source codes of Linux and SPEC2000 benchmarks.

Although implicit learning techniques have been developed for assume-guarantee reasoning and loop invariant inference successfully, challenges still remain. Currently, the learning algorithm is able to infer Boolean functions over tens of Boolean variables. Contextual assumptions over tens of Boolean variables are not enough. Ideally, one would like to have contextual assumptions over hundreds (even thousands) of Boolean variables. On the other hand, it is known that learning arbitrary Boolean functions is infeasible. The scalability of implicit learning techniques cannot be improved satisfactorily by tuning the learning algorithm alone. Combining implicit learning with abstraction will be essential to improve its scalability.

Our second challenge is to extend learning-based techniques to other computation models. In addition to finite automata, probabilistic automata and timed automata are also widely used to specify abstract models. Their verification problems are much more difficult than those for finite automata. Compositional reasoning thus can improve the capacity of model checkers more significantly. Recently, the L^* algorithm is applied in assume-guarantee reasoning for probabilistic automata [46]. The new technique is unfortunately incomplete. Developing a complete learning-based assume-guarantee reasoning technique for probabilistic automata and timed automata will be very useful to their verification.

Through predicate abstraction, learning Boolean functions can be very useful in program analysis. We have successfully applied algorithmic learning to infer both quantified and quantifier-free loop invariants for annotated loops. Applying algorithmic learning to static analysis or program testing will be our last challenge. In the context of program analysis, scalability of the learning algorithm is less of an issue. Formulas over tens of atomic predicates usually suffice to characterize relation among program variables. On the other hand, learning algorithms require oracles to answer queries or generate samples. Designing such oracles necessarily

requires information extracted from program texts. How to extract information will be essential to applying algorithmic learning in static analysis or program testing.

3.3. Decision Procedures

Decision procedures are of utmost importance for us, since they are at the heart of theorem proving and verification. Research in decision procedures started several decades ago, and are now commonly used both in the academia and industry. A decision procedure [55] is an algorithm which returns a correct yes/no answer to a given input decision problem. Many real-world problems can be reduced to the decision problems, making this technique very practical. For example, Intel and AMD are developing solvers for their circuit verification tools, while Microsoft is developing decision procedures for their code analysis tools.

Mathematical logic is the appropriate tool to formulate a decision problem. Most decision problems are formulated as a decidable fragment of a first-order logic interpreted in some specific domain. On such, easy and popular fragment, is propositional (or Boolean) logic, which corresponding decision procedure is called SAT. Representing real problems in SAT often results in awkward encodings that destroy the logical structure of the original problem.

A very popular, effective recent trend is Satisfiability Modulo Theories (SMT) [74], a general technique to solve decision problems formulated as propositional formulas operating on atoms in a given background theory, for example linear real arithmetic. Existing approaches for solving SMT problems can be classified into two categories: *lazy* method [67], and *eager* method [68]. The eager method encodes an SMT problem into an equi-satisfiable SAT problem, while the lazy method employs different theory solvers for each theory and coordinates them appropriately. The eager method does allow the user to express her problem in a natural way, but does not exploit its logical structure to speed up the computation. The lazy approach is more appealing, and has prompted much interest in algorithms for the various background theories important in practice.

Our SMT solver aCiNO is based on the lazy approach. So far, it provides with two (popular) theories only: linear real arithmetic (LRA) and uninterpreted functions (UF). For efficiency consideration, the solver is implemented in an incremental way. It also invokes an online SAT solver, which is now a modified DPLL procedure, so that recovery from conflicts is possible. Our challenge here is twofold: first, to add other theories of interest for the project, we are currently working on fragments of the theory of arrays [61], [34]. The theory of arrays is important because of its use for expressing loop invariants in programs with arrays, but its full first-order theory is undecidable. We are also interested in the theory of bit vectors, very much used for hardware verification.

Theory solvers implement state-of-the-art algorithms which sophistication makes their correct implementation a delicate task. Moreover, SMT solvers themselves employ a quite complex machinery, making them error prone as well [4] We therefore strongly believe that decision procedures, and SMT provers, should come along with a formal assessment of their correctness. As usual, there are two ways: ensure the correctness of an arbitrary output by proving the code, or deliver for each input a certificate ensuring the correctness of the corresponding output when the checker says so. Developing concise certificates together with efficient certificate checkers for the various decision procedures of interest and their combination with SMT is yet another challenge which is at the heart of the project FORMES.

3.4. Simulation

The development of complex embedded systems platforms requires putting together many hardware components, processor cores, application specific co-processors, bus architectures, peripherals, etc. The hardware platform of a project is seldom entirely new. In fact, in most cases, 80 percent of the hardware components are re-used from previous projects or simply are COTS (Commercial Off-The-Shelf) components. There is no need to simulate in great detail these already proven components, whereas there is a need to run fast simulation of the software using these components.

[4] It took almost 20 years to have a correct implementation of a correct version of Shostak's algorithm for combining decision procedures, which can be seen as an ancestor of SMT.

These requirements call for an integrated, modular simulation environment where already proven components can be simulated quickly, (possibly including real hardware in the loop), new components under design can be tested more thoroughly, and the software can be tested on the complete platform with reasonable speed.

Modularity and fast prototyping also have become important aspects of simulation frameworks, for investigating alternative designs with easier re-use and integration of third party components.

The project aims at developing such a rapid prototyping, modular simulation platform, combining new hardware components modeling, verification techniques, fast software simulation for proven components, capable of running the real embedded software application without any change.

To fully simulate a complete hardware platform, one must simulate the processors, the co-processors, together with the peripherals such as network controllers, graphics controllers, USB controllers, etc. A commonly used solution is the combination of some ISS (Instruction Set Simulator) connected to a Hardware Description Language (HDL) simulator which can be implemented by software or by using a FPGA [60] simulator. These solutions tend to present slow iteration design cycles and implementing the FPGA means the hardware has already been designed at low level, which comes normally late in the project and become very costly when using large FPGA platforms. Others have implemented a co-simulation environment, using two separate technologies, typically one using a HDL and another one using an ISS [47], [50], [66]. Some communication and synchronization must be designed and maintained between the two using some inter-process communication (IPC), which slows down the process.

The idea we pursue is to combine hardware modeling and fast simulation into a fully integrated, software based (not using FPGA) simulation environment named SimSoC, which uses a single simulation loop thanks to Transaction Level Modeling (TLM) [36], [23] combined with a new ISS technology designed specifically to fit within the TLM environment.

The most challenging way to enhance simulation speed is to simulate the processors. Processor simulation is achieved with Instruction Set Simulation (ISS). There are several alternatives to achieve such simulation. In *interpretive simulation*, each instruction of the target program is fetched from memory, decoded, and executed. This method is flexible and easy to implement, but the simulation speed is slow as it wastes a lot of time in decoding. Interpretive simulation is used in Simplescalar [35]. Another technique to implement a fast ISS is *dynamic translation* [39], [65], [44] which has been favored by many [63], [44], [64], [65] in the past decade.

With dynamic translation, the binary target instructions are fetched from memory at run-time, like in interpretive simulation. They are decoded on the first execution and the simulator translates these instructions into another representation which is stored into a cache. On further execution of the same instructions, the translated cached version is used. Dynamic translation introduces a translation time phase as part of the overall simulation time. But as the resulting cached code is re-used, the translation time is amortized over time. If the code is modified during run-time, the simulator must invalidate the cached representation. Dynamic translation provides much faster simulation while keeping the advantage of interpretive simulation as it supports the simulation of programs that have either dynamic loading or self-modifying code.

There are many ways of translating binary code into cached data, which each come at a price, with different trade-offs between the translation time and the obtained speed up on cache execution. Also, simulation speed-ups usually don't come for free : most of time there is a trade-off between accuracy and speed.

There are two well known variants of the dynamic translation technology: the target code is translated either directly into machine code for the simulation host, or into an intermediate representation, independent from the host machine, that makes it possible to execute the code with faster speed. Both have pros and cons.

Processor simulation is also achieved in Virtual Machines such as QEMU [28] and GXEMUL [49] that emulate to a large extent the behavior of a particular hardware platform. The technique used in QEMU is a form of dynamic translation. The target code is translated directly into machine code using some pre-determined code patterns that have been pre-compiled with the C compiler. Both QEMU and GXEMUL include many device models of open-source C code, but this code is hard to reuse. The functions that emulate device accesses do not have the same profile. The scheduling process of the parallel hardware entities is not specified well enough to

guarantee the compatibility between several emulators or re-usability of third-party models using the standards from the electronics industry (e.g. IEEE 1666).

A challenge in the development of high performance simulators is to maintain simultaneously fast speed and simulation accuracy. In the FORMES project, we expect to develop a dynamic translation technology satisfying the following additional objectives:

- provide different levels of translation with different degrees of accuracy so that users can choose between accurate and slow (for debugging) or less accurate but fast simulation.
- to take advantage of multi-processor simulation hosts to parallelize the simulation;
- to define intermediate representations of programs that optimize the simulation speed and possibly provide a more convenient format for studying properties of the simulated programs.

Another objective of the FORMES simulation is to extract information from the simulated applications to prove properties. Running a simulation is exercising a test case. In most cases, if a test is failing, a bug has been found. One can use model checking tools to generate tests that can be run on the simulator to check whether the test fails or not on the real application. It is also a goal of FORMES simulation activity to use such formal methods tools to detect bugs, either by generating tests, or by using formal methods tools to analyze the results of simulation sessions.

3.5. Trustworthy Software

Since the early days of software development, computer scientists have been interested in designing methods for improving software quality. Formal methods based on model checking, correctness proofs, common criteria certification, all address this issue in their own way. None of these methods, however, considers the trustworthiness of a given software system as a system-level property, requiring to grasp a given software within its environment of execution.

The major challenge we want to address here is to provide a framework in which to formalize the notion of trustworthiness, to evaluate the trustworthiness of a given software, and if necessary improve it.

To make trustworthiness a fruitful concept, our vision is to formalize it via a hierarchy of observability and controllability degrees: the more the software is observable and controllable, the more its behaviors can be trusted by users. On the other hand, users from different application domains have different expectations from the software they use. For example, aerospace embedded software should be safety-critical while e-commerce software should be insensitive to attacks. As a result, trustworthiness should be domain-specific.

A main challenge is the evaluation of trustworthiness. We believe that users should be responsible for describing the level of trustworthiness they need, in the form of formal requirements that the software should satisfy. A major issue is to come up with some predefined levels of trustworthiness for the major applicative areas. Another is to use stepwise refinement techniques to achieve the appropriate level of trustworthiness. These levels would then drive the design and implementation of a software system: the objective would be to design a model with enough details (observability) to make it possible to check all requirements of that level.

The other challenge is the effective integration of results obtained from different verification methods. There are many verification techniques, like simulation, testing, model checking and theorem proving. These methods may operate on different models of the software to be then executed, while trustworthiness should measure our trust in the real software running in its real execution environment. There are also monitoring and analysis techniques to capture the characteristics of actual executions of the system. Integrating all the analysis in order to decide the trustworthiness level of a software is quite a hard task.

GALLIUM Project-Team

3. Scientific Foundations

3.1. Programming languages: design, formalization, implementation

Like all languages, programming languages are the media by which thoughts (software designs) are communicated (development), acted upon (program execution), and reasoned upon (validation). The choice of adequate programming languages has a tremendous impact on software quality. By "adequate", we mean in particular the following four aspects of programming languages:

- **Safety.** The programming language must not expose error-prone low-level operations (explicit memory deallocation, unchecked array accesses, etc) to the programmers. Further, it should provide constructs for describing data structures, inserting assertions, and expressing invariants within programs. The consistency of these declarations and assertions should be verified through compile-time verification (e.g. static type checking) and run-time checks.
- **Expressiveness.** A programming language should manipulate as directly as possible the concepts and entities of the application domain. In particular, complex, manual encodings of domain notions into programmatic notations should be avoided as much as possible. A typical example of a language feature that increases expressiveness is pattern matching for examination of structured data (as in symbolic programming) and of semi-structured data (as in XML processing). Carried to the extreme, the search for expressiveness leads to domain-specific languages, customized for a specific application area.
- **Modularity and compositionality.** The complexity of large software systems makes it impossible to design and develop them as one, monolithic program. Software decomposition (into semi-independent components) and software composition (of existing or independently-developed components) are therefore crucial. Again, this modular approach can be applied to any programming language, given sufficient fortitude by the programmers, but is much facilitated by adequate linguistic support. In particular, reflecting notions of modularity and software components in the programming language enables compile-time checking of correctness conditions such as type correctness at component boundaries.
- **Formal semantics.** A programming language should fully and formally specify the behaviours of programs using mathematical semantics, as opposed to informal, natural-language specifications. Such a formal semantics is required in order to apply formal methods (program proof, model checking) to programs.

Our research work in language design and implementation centers around the statically-typed functional programming paradigm, which scores high on safety, expressiveness and formal semantics, complemented with full imperative features and objects for additional expressiveness, and modules and classes for compositionality. The OCaml language and system embodies many of our earlier results in this area [36]. Through collaborations, we also gained experience with several domain-specific languages based on a functional core, including XML processing (XDuce, CDuce), reactive functional programming, distributed programming (JoCaml), and hardware modeling (ReFLect).

3.2. Type systems

Type systems [49] are a very effective way to improve programming language reliability. By grouping the data manipulated by the program into classes called types, and ensuring that operations are never applied to types over which they are not defined (e.g. accessing an integer as if it were an array, or calling a string as if it were a function), a tremendous number of programming errors can be detected and avoided, ranging from the trivial (misspelled identifier) to the fairly subtle (violation of data structure invariants). These restrictions are also very effective at thwarting basic attacks on security vulnerabilities such as buffer overflows.

The enforcement of such typing restrictions is called type checking, and can be performed either dynamically (through run-time type tests) or statically (at compile-time, through static program analysis). We favor static type checking, as it catches bugs earlier and even in rarely-executed parts of the program, but note that not all type constraints can be checked statically if static type checking is to remain decidable (i.e. not degenerate into full program proof). Therefore, all typed languages combine static and dynamic type-checking in various proportions.

Static type checking amounts to an automatic proof of partial correctness of the programs that pass the compiler. The two key words here are *partial*, since only type safety guarantees are established, not full correctness; and *automatic*, since the proof is performed entirely by machine, without manual assistance from the programmer (beyond a few, easy type declarations in the source). Static type checking can therefore be viewed as the poor man's formal methods: the guarantees it gives are much weaker than full formal verification, but it is much more acceptable to the general population of programmers.

3.2.1. Type systems and language design.

Unlike most other uses of static program analysis, static type-checking rejects programs that it cannot analyze safe. Consequently, the type system is an integral part of the language design, as it determines which programs are acceptable and which are not. Modern typed languages go one step further: most of the language design is determined by the *type structure* (type algebra and typing rules) of the language and intended application area. This is apparent, for instance, in the XDuce and CDuce domain-specific languages for XML transformations [46], [42], whose design is driven by the idea of regular expression types that enforce DTDs at compile-time. For this reason, research on type systems – their design, their proof of semantic correctness (type safety), the development and proof of associated type checking and inference algorithms – plays a large and central role in the field of programming language research, as evidenced by the huge number of type systems papers in conferences such as Principles of Programming Languages.

3.2.2. Polymorphism in type systems.

There exists a fundamental tension in the field of type systems that drives much of the research in this area. On the one hand, the desire to catch as many programming errors as possible leads to type systems that reject more programs, by enforcing fine distinctions between related data structures (say, sorted arrays and general arrays). The downside is that code reuse becomes harder: conceptually identical operations must be implemented several times (say, copying a general array and a sorted array). On the other hand, the desire to support code reuse and to increase expressiveness leads to type systems that accept more programs, by assigning a common type to broadly similar objects (for instance, the `Object` type of all class instances in Java). The downside is a loss of precision in static typing, requiring more dynamic type checks (downcasts in Java) and catching fewer bugs at compile-time.

Polymorphic type systems offer a way out of this dilemma by combining precise, descriptive types (to catch more errors statically) with the ability to abstract over their differences in pieces of reusable, generic code that is concerned only with their commonalities. The paradigmatic example is parametric polymorphism, which is at the heart of all typed functional programming languages. Many forms of polymorphic typing have been studied since then. Taking examples from our group, the work of Rémy, Vouillon and Garrigue on row polymorphism [53], integrated in OCaml, extended the benefits of this approach (reusable code with no loss of typing precision) to object-oriented programming, extensible records and extensible variants. Another example is the work by Pottier on subtype polymorphism, using a constraint-based formulation of the type system [50].

3.2.3. Type inference.

Another crucial issue in type systems research is the issue of type inference: how many type annotations must be provided by the programmer, and how many can be inferred (reconstructed) automatically by the typechecker? Too many annotations make the language more verbose and bother the programmer with unnecessary details. Too few annotations make type checking undecidable, possibly requiring heuristics, which is unsatisfactory. OCaml requires explicit type information at data type declarations and at component interfaces, but infers all other types.

In order to be predictable, a type inference algorithm must be complete. That is, it must not find *one*, but *all* ways of filling in the missing type annotations to form an explicitly typed program. This task is made easier when all possible solutions to a type inference problem are *instances* of a single, *principal* solution.

Maybe surprisingly, the strong requirements – such as the existence of principal types – that are imposed on type systems by the desire to perform type inference sometimes lead to better designs. An illustration of this is row variables. The development of row variables was prompted by type inference for operations on records. Indeed, previous approaches were based on subtyping and did not easily support type inference. Row variables have proved simpler than structural subtyping and more adequate for typechecking record update, record extension, and objects.

Type inference encourages abstraction and code reuse. A programmer's understanding of his own program is often initially limited to a particular context, where types are more specific than strictly required. Type inference can reveal the additional generality, which allows making the code more abstract and thus more reuseable.

3.3. Compilation

Compilation is the automatic translation of high-level programming languages, understandable by humans, to lower-level languages, often executable directly by hardware. It is an essential step in the efficient execution, and therefore in the adoption, of high-level languages. Compilation is at the interface between programming languages and computer architecture, and because of this position has had considerable influence on the designs of both. Compilers have also attracted considerable research interest as the oldest instance of symbolic processing on computers.

Compilation has been the topic of much research work in the last 40 years, focusing mostly on high-performance execution ("optimization") of low-level languages such as Fortran and C. Two major results came out of these efforts: one is a superb body of performance optimization algorithms, techniques and methodologies; the other is the whole field of static program analysis, which now serves not only to increase performance but also to increase reliability, through automatic detection of bugs and establishment of safety properties. The work on compilation carried out in the Gallium group focuses on a less investigated topic: compiler certification.

3.3.1. Formal verification of compiler correctness.

While the algorithmic aspects of compilation (termination and complexity) have been well studied, its semantic correctness – the fact that the compiler preserves the meaning of programs – is generally taken for granted. In other terms, the correctness of compilers is generally established only through testing. This is adequate for compiling low-assurance software, themselves validated only by testing: what is tested is the executable code produced by the compiler, therefore compiler bugs are detected along with application bugs. This is not adequate for high-assurance, critical software which must be validated using formal methods: what is formally verified is the source code of the application; bugs in the compiler used to turn the source into the final executable can invalidate the guarantees so painfully obtained by formal verification of the source.

To establish strong guarantees that the compiler can be trusted not to change the behavior of the program, it is necessary to apply formal methods to the compiler itself. Several approaches in this direction have been investigated, including translation validation, proof-carrying code, and type-preserving compilation. The approach that we currently investigate, called *compiler verification*, applies program proof techniques to the compiler itself, seen as a program in particular, and use a theorem prover (the Coq system) to prove that the generated code is observationally equivalent to the source code. Besides its potential impact on the critical software industry, this line of work is also scientifically fertile: it improves our semantic understanding of compiler intermediate languages, static analyses and code transformations.

3.4. Interface with formal methods

Formal methods refer collectively to the mathematical specification of software or hardware systems and to the verification of these systems against these specifications using computer assistance: model checkers, theorem provers, program analyzers, etc. Despite their costs, formal methods are gaining acceptance in the critical software industry, as they are the only way to reach the required levels of software assurance.

In contrast with several other Inria projects, our research objectives are not fully centered around formal methods. However, our research intersects formal methods in the following two areas, mostly related to program proofs using proof assistants and theorem provers.

3.4.1. Software-proof codesign

The current industrial practice is to write programs first, then formally verify them later, often at huge costs. In contrast, we advocate a codesign approach where the program and its proof of correctness are developed in interaction, and are interested in developing ways and means to facilitate this approach. One possibility that we currently investigate is to extend functional programming languages such as Caml with the ability to state logical invariants over data structures and pre- and post-conditions over functions, and interface with automatic or interactive provers to verify that these specifications are satisfied. Another approach that we practice is to start with a proof assistant such as Coq and improve its capabilities for programming directly within Coq. Finally, we also participate in the FoCaLiZe project, which designs and implements an environment for combined programming and proving [23] [52].

3.4.2. Mechanized specifications and proofs for programming languages components

We emphasize mathematical specifications and proofs of correctness for key language components such as semantics, type systems, type inference algorithms, compilers and static analyzers. These components are getting so large that machine assistance becomes necessary to conduct these mathematical investigations. We have already mentioned using proof assistants to verify compiler correctness. We are also interested in using them to specify and reason about semantics and type systems. These efforts are part of a more general research topic that is gaining importance: the formal verification of the tools that participate in the construction and certification of high-assurance software.

MUTANT Project-Team

3. Scientific Foundations

3.1. Real-time Machine Listening

When human listeners are confronted with musical sounds, they rapidly and automatically find their way in the music. Even musically untrained listeners have an exceptional ability to make rapid judgments about music from short examples, such as determining music style, performer, beating, and specific events such as instruments or pitches. Making computer systems capable of similar capabilities requires advances in both music cognition, and analysis and retrieval systems employing signal processing and machine learning.

In a panel session at the 13th National Conference on Artificial Intelligence in 1996, Rodney Brooks (noted figure in robotics) remarked that while automatic speech recognition was a highly researched domain, there had been few works trying to build machines able to understand "non-speech sound". He went further to name this as one of the biggest challenges faced by Artificial Intelligence [41]. More than 15 years have passed. Systems now exist that are able to analyze the contents of music and audio signals and communities such as International Symposium on Music Information Retrieval (MIR) and Sound and Music Computing (SMC) have formed. But we still lack reliable Real-Time machine listening systems.

The first thorough study of machine listening appeared in Eric Scheirer's PhD thesis at MIT Media Lab in 2001 [40] with a focus on low-level listening such as pitch and musical tempo, paving the way for a decade of research. Since the work of Scheirer, the literature has focused on task-dependent methods for machine listening such as pitch estimation, beat detection, structure discovery and more. Unfortunately, the majority of existing approaches are designed for information retrieval on large databases or off-line methods. Whereas the very act of listening is real-time, very little literature exists for supporting real-time machine listening. This argument becomes more clear while looking at the yearly Music Information Retrieval Evaluation eXchange (MIREX), with different retrieval tasks and submitted systems from international institutions, where almost no emphasis exists on real-time machine listening. Most MIR contributions focus on off-line approaches to information retrieval (where the system has access to future data) with less focus on on-line and realtime approaches to information decoding.

On another front, most MIR algorithms suffer from modeling of temporal structures and temporal dynamics specific to music (where most algorithms have roots in speech or biological sequence without correct adoption to temporal streams such as music). Despite tremendous progress using modern signal processing and statistical learning, there is much to be done to achieve the same level of abstract understanding for example in text and image analysis on music data. On another hand, it is important to notice that even untrained listeners are easily able to capture many aspects of formal and symbolic structures from an audio stream in realtime. Realtime machine listening is thus still a major challenge for artificial sciences that should be addressed both on application and theoretical fronts.

In the MUTANT project, we focus on realtime and online methods of music information retrieval out of audio signals. One of the primary goals of such systems is to fill in the gap between *signal representation* and *symbolic information* (such as pitch, tempo, expressivity, etc.) contained in music signals. MUTANT's current activities focus on two main applications: *score following* or realtime audio-to-score alignment [2], and realtime transcription of music signals [20] with impacts both on signal processing using machine learning techniques and their application in real-world scenarios.

The team-project will focus on two aspects of realtime machine listening:

1. **Application-Driven Approach:** First, to enhance and foster existing application-driven approaches within the team such as realtime alignment algorithms and polyphonic pitch transcription. Our contributions on this line correspond to extensions of existing algorithmic approaches to realtime audio alignment and transcription to create new interactive application paradigms with new algorithmic

approaches. Arshia Cont's ongoing realtime alignment in *Antescofo* as well as realtime transcription using non-negative factorization methods [20] are examples of this.

2. **Music Information Geometry:** In parallel to concrete applications, we hope to theoretically contribute to the problem of signal representations of audio streams for effortless retrieval of high-level information structures. We have recently shown in [4] that the gap between the symbolic/semantic and signal aspects of music information mostly lies on constructing a well-behaved representational space before any algorithmic considerations, by employing the emerging methods of *information geometry*. Arnaud Dessein's ongoing PhD thesis is focused on this aspect of the project.

3.2. Synchronous and realtime programming for computer music

The second aspect of an interactive music system is to *react* to extracted high-level and low-level music information based on pre-defined actions. The simplest scenario is *automatic accompaniment*, delegating the interpretation of one or several musical voices to a computer, in interaction with a live solo (or ensemble) musician(s). The most popular form of such systems is the automatic accompaniment of an orchestral recording with that of a soloist in the classical music repertoire (concertos for example). In the larger context of interactive music systems, the "notes" or musical elements in the accompaniment are replaced by "programs" that are written during the phase of composition and are evaluated in realtime in reaction and relative to musicians' performance. The programs in question here can range from sound playback, to realtime sound synthesis by simulating physical models, and realtime transformation of musician's audio and gesture.

Such musical practice is commonly referred to as the *realtime school* in computer music, developed naturally with the invention of the first score following systems, and led to the invention of the first prototype of realtime digital signal processors [28] and subsequents [31], and the realtime graphical programming environment *Max* for their control [37] at Ircam. With the advent and availability of DSPs in personal computers, integrated realtime event and signal processing graphical language *MaxMSP* was developed [38] at Ircam, which today is the worldwide standard platform for realtime interactive arts programming. This approach to music making was first formalized by composers such as Philippe Manoury and Pierre Boulez, in collaboration with researchers at Ircam, and soon became a standard in musical composition with computers.

Besides realtime performance and implementation issues, little work has underlined the formal aspects of such practices in realtime music programming, in accordance to the long and quite rich tradition of musical notations. Recent progress has convinced both the researcher and artistic bodies that this programming paradigm is close to *synchronous reactive programming languages*, with concrete analogies between both: parallel synchrony and concurrency is equivalent to musical polyphony, periodic sampling to rhythmic patterns, hierarchical structures to micro-polyphonies, and demands for novel hybrid models of time among others. *Antescofo* is therefore an early response to such demands that needs further explorations and studies.

Within the MUTANT project, we propose to tackle this aspect of the research within three consecutive lines:

- **Development of a Synchronous DSL for Real Time Musician-Computer Interaction:** Ongoing and continuous extensions of the *Antescofo* language following user requests and by inscribing them within a coherent framework for the handling of temporal musical relationships. José Echeveste's ongoing PhD thesis focuses on the research and development of these aspects. Recent formalizations of the *Antescofo* language has been published in [6].
- **Formal Methods:** Failure during an artistic performance should be avoided. This naturally leads to the use of formal methods, like static analysis or model checking, to ensure formally that the execution of an *Antescofo* program will satisfy some expected property. The checked properties may also provide some assistance to the composer especially in the context of "non deterministic score" in an interactive framework.

3.3. Off-the-shelf Operating Systems for Real-time Audio

While operating systems shield the computer hardware from all other software, it provides a comfortable environment for program execution and evades offensive use of hardware by providing various services related

to essential tasks. However, integrating discrete and continuous multimedia data demands additional services, especially for real-time processing of continuous-media such as audio and video. To this end interactive systems are sometimes referred to as off-the-shelf operating systems for real-time audio. The difficulty in providing correct real-time services has much to do with human perception. Correctness for real-time audio is more stringent than video because human ear is more sensitive to audio gaps and glitches than human eye is to video jitter [43]. Here we expose the foundations of existing sound and music operating systems and focus on their major drawbacks with regards to today practices.

An important aspect of any real-time operating system is fault-tolerance with regards to short-time failure of continuous-media computation, delivery delay or missing deadlines. Existing multimedia operating systems are soft real-time where missing a deadline does not necessarily lead to system failure and have their roots in pioneering work in [42]. Soft real-time is acceptable in simple applications such as video-on-demand delivery, where initial delay in delivery will not directly lead to critical consequences and can be compensated (general scheme used for audio-video synchronization), but with considerable consequences for Interactive Systems: Timing failure in interactive systems will heavily affect inter-operability of models of computation, where incorrect ordering can lead to unpredictable and unreliable results. Moreover, interaction between computing and listening machines (both dynamic with respect of internal computation and physical environment) requires tighter and explicit temporal semantics since interaction between physical environment and the system can be continuous and not demand-driven.

Fulfilling timing requirements of continuous media demands explicit use of scheduling techniques. As shown earlier, existing Interactive Music Systems rely on combined event/signal processing. In real-time, scheduling techniques aim at gluing the two engines together with the aim of timely delivery of computations between agents and components, from the physical environment, as well as to hardware components. The first remark in studying existing system is that they all employ static scheduling, whereas interactive computing demands more and more time-aware and context-aware dynamic methods. The scheduling mechanisms are neither aware of time, nor the nature and semantics of computations at stake. Computational elements are considered in a functional manner and reaction and execution requirements are simply ignored. For example, *Max* scheduling mechanisms can delay message delivery when many time-critical tasks are requested within one cycle [38]. *SuperCollider* uses Earliest-Deadline-First (EDF) algorithms and cycles can be simply missed [35]. This situation leads to non-deterministic behavior with deterministic components and poses great difficulties for preservation of underlying techniques, art pieces, and algorithms. The situation has become worse with the demand for nomad physical computing where individual programs and modules are available but no action coordination or orchestration is proposed to design integrated systems. System designers are penalized for expressivity, predictability and reliability of their design despite potentially reliable components.

Existing systems have been successful in programing and executing small system comprised of few programs. However, severe problems arise when scaling from program to system-level for moderate or complex programs leading to unpredictable behavior. Computational elements are considered as functions and reaction and execution requirements are simply ignored. System designers have uniformly chosen to hide timing properties from higher abstractions, and despite its utmost importance in multimedia computing, timing becomes an accident of implementation. This confusing situation for both artists and system designers, is quite similar to the one described in Dr. Edward Lee's seminal paper "Computing needs time" stating: "general-purpose computers are increasingly asked to interact with physical processes through integrated media such as audio. [...] and they don't always do it well. The technological basis that engineers have chosen for general-purpose computing [...] does not support these applications well. Changes that ensure this support could improve them and enable many others" [30].

Despite all shortcomings, one of the main advantages of environments such as *Max* and *PureData* to other available systems, and probably the key to their success, is their ability to handle both synchronous processes (such as audio or video delivery and processing) within an asynchronous environment (user and environmental interactions). Besides this fact, multimedia service scheduling at large has a tendency to go more and more towards computing besides mere on-time delivery. This brings in the important question of hybrid scheduling of heterogeneous time and computing models in such environments, a subject that has had very few studies

in multimedia processing but studied in areas such simulation applications. We hope to address this issue scientifically by first an explicit study of current challenges in the domain, and second by proposing appropriate methods for such systems. This research is inscribed in the three year ANR project INEDIT coordinated by the team leader (started in September 2012).

PARKAS Project-Team

3. Scientific Foundations

3.1. Presentation and originality of the PARKAS team

Our project is founded on our expertise in three complementary domains: (1) synchronous functional programming and its extensions to deal with features such as communication with bounded buffers and dynamic process creation; (2) mathematical models for synchronous circuits; (3) compilation techniques for synchronous languages and optimizing/parallelizing compilers.

A strong point of the team is its experience and investment in the development of languages and compilers. Members of the team also have direct collaborations for several years with major industrial companies in the field and several of our results are integrated in successful products. Our main results are briefly summarized below.

3.1.1. Synchronous functional programming

In [19], Paul Caspi and Marc Pouzet introduced *synchronous Kahn networks* as those Kahn networks that can be statically scheduled and executed with bounded buffers. This was the origin of the language LUCID SYNCHRONE, [1][2] an ML extension of the synchronous language LUSTRE with higher-order features, dedicated type systems (clock calculus as a type system [19], [29], initialization analysis [30] and causality analysis [31]). The language integrates original features that are not found in other synchronous languages: such as combinations of data flow, control flow, hierarchical automata and signals [28], [27], and modular code generation [20], [17].

In 2000, Marc Pouzet started to collaborate with the SCADE team of Esterel-Technologies on the design of a new version of SCADE. [3] Several features of LUCID SYNCHRONE are now integrated into SCADE 6, which has been distributed since 2008, including the programming constructs `merge`, `reset`, the clock calculus and the type system. Several results have been developed jointly with Jean-Louis Colaço and Bruno Pagano from Esterel-Technologies, such as ways of combining data-flow and hierarchical automata, and techniques for their compilation, initialization analysis, etc.

Dassault-Systèmes (Grenoble R&D center, part of Delmia-automation) developed the language LCM, a variant of LUCID SYNCHRONE that is used for the simulation of factories. LCM follows closely the principles and programming constructs of LUCID SYNCHRONE (higher-order, type inference, mix of data-flow and hierarchical automata). The team in Grenoble is integrating this development into a new compiler for the language Modelica. [4]

In parallel, the goal of REACTIVEML[5] was to integrate a synchronous concurrency model into an existing ML language, with no restrictions on expressiveness, so as to program a large class of reactive systems, including efficient simulations of millions of communicating processes (e.g., sensor networks), video games with many interactions, physical simulations, etc. For such applications, the synchronous model simplifies system design and implementation, but the expressiveness of the algorithmic part of the language is just as essential, as is the ability to create or stop a process dynamically.

The development of REACTIVEML was started by Louis Mandel during his PhD thesis [42], [38] and is ongoing. The language extends OCAML[6] with Esterel-like synchronous primitives — synchronous composition, broadcast communication, pre-emption/suspension — applying the solution of Boussinot [18] to solve causality issues.

[1] http://www.di.ens.fr/~pouzet/lucid-synchrone
[2] The name is a reference to Lustre which stands for "Lucid Synchrone et Temps réel".
[3] http://www.esterel-technologies.com/products/scade-suite/
[4] http://www.3ds.com/products/catia/portfolio/dymola/overview/
[5] http://rml.lri.fr/
[6] More precisely a subset of OCAML without objects or functors.

Several open problems have been solved by Louis Mandel: the interaction between ML features (higher-order) and reactive constructs with a proper type system; efficient simulation that avoids busy waiting. The latter problem is particularly difficult in synchronous languages because of possible reactions to the absence of a signal. In the REACTIVEML implementation, there is no busy waiting: inactive processes have no impact on the overall performance. It turns out that this enables REACTIVEML to simulate millions of (logical) parallel processes and to compete with the best event-driven simulators [43].

REACTIVEML has been used for simulating routing protocols in ad-hoc networks [37] and large scale sensor networks [53]. The designer benefits from a real programming language that gives precise control of the level of simulation (e.g., each network layer up to the MAC layer) and programs can be connected to models of the physical environment programmed with LUTIN [52]. REACTIVEML is used since 2006 by the synchronous team at VERIMAG, Grenoble (in collaboration with France-Telecom) for the development of low-consumption routing protocols in sensor networks.

3.1.2. Relaxing synchrony with buffer communication

In the data-flow synchronous model, the clock calculus is a static analysis that ensures execution in bounded memory. It checks that the values produced by a node are instantaneously consumed by connected nodes (synchronous constraint). To program Kahn process networks with bounded buffers (as in video applications), it is thus necessary to explicitly place nodes that implement buffers. The buffers sizes and the clocks at which data must be read or written have to be computed manually. In practice, it is done with simulation or successive tries and errors. This task is difficult and error prone. The aim of the n-synchronous model is to automatically compute at compile time these values while insuring the absence of deadlock.

Technically, it allows processes to be composed whenever they can be synchronized through a bounded buffer [21], [22]. The new flexibility is obtained by relaxing the clock calculus by replacing the equality of clocks by a sub-typing rule. The result is a more expressive language which still offers the same guarantees as the original. The first version of the model was based on clocks represented as ultimately periodic binary words [57]. It was algorithmically expensive and limited to periodic systems. In [25], an abstraction mechanism is proposed which permits direct reasoning on sets of clocks that are defined as a rational slope and two shifts. An implementation of the n-synchronous model, named LUCY-N, was developed in 2009 [39], as was a formalization of the theory in COQ [26]. We also worked on low-level compiler and runtime support to parallelize the execution of relaxed synchronous systems, proposing a portable intermediate language and runtime library called ERBIUM [44].

This work started as a collaboration between Marc Pouzet (LIP6, Paris, then LRI and Inria Proval, Orsay), Marc Duranton (Philips Research then NXP, Eindhoven), Albert Cohen (Inria Alchemy, Orsay) and Christine Eisenbeis (Inria Alchemy, Orsay) on the real-time programming of video stream applications in set-top boxes. It was significantly extended by Louis Mandel and Florence Plateau during her PhD thesis [47] (supervised by Marc Pouzet and Louis Mandel). Low-level support has been investigated with Cupertino Miranda, Philippe Dumont (Inria Alchemy, Orsay) and Antoniu Pop (Mines ParisTech).

3.1.3. Polyhedral compilation and optimizing compilers

Despite decades of progress, the best parallelizing and optimizing compilers still fail to extract parallelism and to perform the necessary optimizations to harness multi-core processors and their complex memory hierarchies. *Polyhedral compilation* aims at facilitating the construction of more effective optimization and parallelization algorithms. It captures the flow of data between individual instances of statements in a loop nest, allowing to accurately model the behavior of the program and represent complex parallelizing and optimizing transformations. Affine multidimensional scheduling is one of the main tools in polyhedral compilation [32]. Albert Cohen, in collaboration with Cédric Bastoul, Sylvain Girbal, Nicolas Vasilache, Louis-Noël Pouchet and Konrad Trifunovic (LRI and Inria Alchemy, Orsay) has contributed to a large number of research, development and transfer activities in this area.

The relation between polyhedral compilation and data-flow synchrony has been identified through data-flow array languages [36], [35], [54], [33] and the study of the scheduling and mapping algorithms for these

languages. We would like to deepen the exploration of this link, embedding polyhedral techniques into the compilation flow of data-flow, relaxed synchronous languages.

Our previous work led to the design of a theoretical and algorithmic framework rooted in the polyhedral model of compilation, and to the implementation of a set of tools based on production compilers (Open64, GCC) and source-to-source prototypes (PoCC, http://pocc.sourceforge.net). We have shown that not only does this framework simplify the problem of building complex loop nest optimizations, but also that it scales to real-world benchmarks [23], [34], [50], [49]. The polyhedral model has finally evolved into a mature, production-ready approach to solve the challenges of maximizing the scalability and efficiency of loop-based computations on a variety of high performance and embedded targets.

After an initial experiment with Open64 [24], [23], we ported these techniques to the GCC compiler [48], [56], [55], applying them to multi-level parallelization and optimization problems, including vectorization and exploitation of thread-level parallelism. Independently, we made significant progress in the design of effective optimization heuristics, working on the interactions between the semantics of the compiler's intermediate representation and the structure of the optimization space [50], [49], [51]. These results open opportunities for complex optimizations that target larger problems, such as the scheduling and placement of process networks, or the offloading of computational kernels to hardware accelerators (such as GPUs).

3.1.4. Automatic compilation of high performance circuits

For both cost and performance reasons, computing systems tightly couple parts realized in hardware with parts realized in software. The boundary between hardware and software keeps moving with the underlying technology and the external economic pressure. Moreover, thanks to FPGA technology, hardware itself has become programmable. There is now a pressing need from industry for hardware/software co-design, and for tools which automatically turn software code into hardware circuits, or more usually, into hybrid code that simultaneously targets GPUs, multiple cores, encryption ASICs, and other specialized chips.

Departing from customary C-to-VHDL compilation, we trust that sharper results can be achieved from source programs that specify bit-wise time/space behavior in a rigorous synchronous language, rather than just the I/O behavior in some (ill-specified) subset of C. This specification allows the designer to also program the (asynchronous) environment in which to operate the entire system, and to profile/measure/control each variable of the design.

At any time, the designer can edit a single specification of the system, from which both the software and the hardware are automatically compiled, and guaranteed to be compatible. Once correct (functionally and with respect to the behavioral specification), the application can be automatically deployed (and tested) on a hard/soft hybrid co-design support.

Key aspects of the advocated methodology were validated by Jean Vuillemin in the design of a PAL2HDTV video sampler [45], [46]. The circuit was automatically compiled from a synchronous source specification, decorated and guided by a few key hints to the hardware back-end, that targetted an FPGA running at real-time video specifications: a tightly-packed highly-efficient design at 240MHz, generated 100% automatically from the application specification source code, and including all run-time/debug/test/validate ancillary software. It was subsequently commercialized on FPGA by LetItWave, and then on ASIC by Zoran. This successful experience underlines our research perspectives on parallel synchronous programming.

PI.R2 Project-Team

3. Scientific Foundations

3.1. Proof theory and the Curry-Howard correspondence

3.1.1. Proofs as programs

Proof theory is the branch of logic devoted to the study of the structure of proofs. An essential contributor to this field is Gentzen [49] who developed in 1935 two logical formalisms that are now central to the study of proofs. These are the so-called "natural deduction", a syntax that is particularly well-suited to simulate the intuitive notion of reasoning, and the so-called "sequent calculus", a syntax with deep geometric properties that is particularly well-suited for proof automation.

Proof theory gained a remarkable importance in computer science when it became clear, after genuine observations first by Curry in 1958 [43], then by Howard and de Bruijn at the end of the 60's [56], [73], that proofs had the very same structure as programs: for instance, natural deduction proofs can be identified as typed programs of the ideal programming language known as λ-calculus.

This proofs-as-programs correspondence has been the starting point to a large spectrum of researches and results contributing to deeply connect logic and computer science. In particular, it is from this line of work that Coquand's Calculus of Constructions [40] stemmed out – a formalism that is both a logic and a programming language and that is at the source of the Coq system [39].

3.1.2. Towards the calculus of constructions

The λ-calculus, defined by Church [38], is a remarkably succinct model of computation that is defined via only three constructions (abstraction of a program with respect to one of its parameters, reference to such a parameter, application of a program to an argument) and one reduction rule (substitution of the formal parameter of a program by its effective argument). The λ-calculus, which is Turing-complete, i.e. which has the same expressiveness as a Turing machine (there is for instance an encoding of numbers as functions in λ-calculus), comes with two possible semantics referred to as call-by-name and call-by-value evaluations. Of these two semantics, the first one, which is the simplest to characterise, has been deeply studied in the last decades [34].

For explaining the Curry-Howard correspondence, it is important to distinguish between intuitionistic and classical logic: following Brouwer at the beginning of the 20^{th} century, classical logic is a logic that accepts the use of reasoning by contradiction while intuitionistic logic proscribes it. Then, Howard's observation is that the proofs of the intuitionistic natural deduction formalism exactly coincide with programs in the (simply typed) λ-calculus.

A major achievement has been accomplished by Martin-Löf who designed in 1971 a formalism, referred to as modern type theory, that was both a logical system and a (typed) programming language [62].

In 1985, Coquand and Huet [40], [41] in the Formel team of Inria-Rocquencourt explored an alternative approach based on Girard-Reynolds' system F [50], [68]. This formalism, called the Calculus of Constructions, served as logical foundation of the first implementation of Coq in 1984. Coq was called CoC at this time.

3.1.3. The Calculus of Inductive Constructions

The first public release of CoC dates back to 1989. The same project-team developed the programming language Caml (nowadays coordinated by the Gallium team) that provided the expressive and powerful concept of algebraic data types (a paragon of it being the type of list). In CoC, it was possible to simulate algebraic data types, but only through a not-so-natural not-so-convenient encoding.

In 1989, Coquand and Paulin [42] designed an extension of the Calculus of Constructions with a generalisation of algebraic types called inductive types, leading to the Calculus of Inductive Constructions (CIC) that started to serve as a new foundation for the Coq system. This new system, which got its current definitive name Coq, was released in 1991.

In practice, the Calculus of Inductive Constructions derives its strength from being both a logic powerful enough to formalise all common mathematics (as set theory is) and an expressive richly-typed functional programming language (like ML but with a richer type system, no effects and no non-terminating functions).

3.2. The development of Coq

Since 1984, about 40 persons have contributed to the development of Coq, out of which 7 persons have contributed to bring the system to the place it is now. First Thierry Coquand through his foundational theoretical ideas, then Gérard Huet who developed the first prototypes with Thierry Coquand and who headed the Coq group until 1998, then Christine Paulin who was the main actor of the system based on the CIC and who headed the development group from 1998 to 2006. On the programming side, important steps were made by Chet Murthy who raised Coq from the prototypical state to a reasonably scalable system, Jean-Christophe Filliâtre who turned to concrete the concept of a small trustful certification kernel on which an arbitrary large system can be set up, Bruno Barras and Hugo Herbelin who, among other extensions, reorganised Coq on a new smoother and more uniform basis able to support a new round of extensions for the next decade.

The development started from the Formel team at Rocquencourt but, after Christine Paulin got a position in Lyon, it spread to École Normale Supérieure de Lyon. Then, the task force there globally moved to the University of Orsay when Christine Paulin got a new position there. On the Rocquencourt side, the part of Formel involved in ML moved to the Cristal team (now Gallium) and Formel got renamed into Coq. Gérard Huet left the team and Christine Paulin started to head a Coq team bilocalised at Rocquencourt and Orsay. Gilles Dowek became the head of the team which was renamed into LogiCal. Following Gilles Dowek who got a position at École Polytechnique, LogiCal globally moved to Futurs with a bilocalisation on Orsay and Palaiseau. It then split again giving birth to ProVal. At the same time, the Marelle team (formerly Lemme, formerly Croap) which has been a long partner of the Formel team, invested more and more energy in both the formalisation of mathematics in Coq and in user interfaces for Coq.

After various other spreadings resulting from where the wind pushed former PhD students, the development of Coq got multi-site with the development now realised by employees of Inria, the CNAM and Paris 7.

We next briefly describe the main components of Coq.

3.2.1. The underlying logic and the verification kernel

The architecture adopts the so-called de Bruijn principle: the well-delimited *kernel* of Coq ensures the correctness of the proofs validated by the system. The kernel is rather stable with modifications tied to the evolution of the underlying Calculus of Inductive Constructions formalism. The kernel includes an interpreter of the programs expressible in the CIC and this interpreter exists in two flavours: a customisable lazy evaluation machine written in OCaml and a call-by-value bytecode interpreter written in C dedicated to efficient computations. The kernel also provides a module system.

3.2.2. Programming and specification languages

The concrete user language of Coq, called *Gallina*, is a high-level language built on top of the CIC. It includes a type inference algorithm, definitions by complex pattern-matching, implicit arguments, mathematical notations and various other high-level language features. This high-level language serves both for the development of programs and for the formalisation of mathematical theories. Coq also provides a large set of commands. Gallina and the commands together forms the *Vernacular* language of Coq.

3.2.3. Libraries

Libraries are written in the vernacular language of Coq. There are libraries for various arithmetical structures and various implementations of numbers (Peano numbers, implementation of $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$ with binary digits, implementation of $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$ using machine words, axiomatisation of $\mathbb{R}$). There are libraries for lists, list of a specified length, sorts, and for various implementations of finite maps and finite sets. There are libraries on relations, sets, orders.

3.2.4. Tactics

The tactics are the methods available to conduct proofs. This includes the basic inference rules of the CIC, various advanced higher level inference rules and all the automation tactics. Regarding automation, there are tactics for solving systems of equations, for simplifying ring or field expressions, for arbitrary proof search, for semi-decidability of first-order logic and so on. There is also a powerful and popular untyped scripting language for combining tactics into more complex tactics.

Note that all tactics of Coq produce proof certificates that are checked by the kernel of Coq. As a consequence, possible bugs in proof methods do not hinder the confidence in the correctness of the Coq checker. Note also that the CIC being a programming language, tactics can be written (and certified) in the own language of Coq if needed.

3.2.5. Extraction

Extraction is a component of Coq that maps programs (or even computational proofs) of the CIC to functional programs (in OCaml, Scheme or Haskell). Especially, a program certified by Coq can further be extracted to a program of a full-fledged programming language then benefiting of the efficient compilation, linking tools, profiling tools, ... of the target software.

3.3. Dependently typed programming languages

Dependently typed programming (shortly DTP) is an emerging concept referring to the diffuse and broadening tendency to develop programming languages with type systems able to express program properties finer than the usual information of simply belonging to specific data-types. The type systems of dependently-typed programming languages allow to express properties *dependent* of the input and the output of the program (for instance that a sorting program returns a list of same size as its argument). Typical examples of such languages were the Cayenne language, developed in the late 90's at Chalmers University in Sweden and the DML language developed at Boston. Since then, various new tools have been proposed, either as typed programming languages whose types embed equalities (Ωmega at Portland, ATS at Boston, ...) or as hybrid logic/programming frameworks (Agda at Chalmers University, Twelf at Carnegie, Delphin at Yale, OpTT at U. Iowa, Epigram at Nottingham, ...).

DTP contributes to a general movement leading to the fusion between logic and programming. Coq, whose language is both a logic and a programming language which moreover can be extracted to pure ML code plays a role in this movement and some frameworks for DTP have been proposed on top of Coq (Concoqtion at Rice and Colorado, Ynot at Harvard, Why in the ProVal team at Inria). It also connects to Hoare logic, providing frameworks where pre- and post-conditions of programs are tied with the programs.

DTP approached from the programming language side generally benefits of a full-fledged language (e.g. supporting effects) with efficient compilation. DTP approached from the logic side generally benefits of an expressive specification logic and of proof methods so as to certify the specifications. The weakness of the approach from logic however is generally the weak support for effects or partial functions.

3.3.1. Type-checking and proof automation

In between the decidable type systems of conventional data-types based programming languages and the full expressiveness of logically undecidable formulae an active field of research explores a spectrum of decidable or semi-decidable type systems for possible use in dependently programming languages. At the beginning of the spectrum, this includes for instance the system F's extension ML_F of the ML type system or the generalisation

of abstract data types with type constraints (G.A.D.T.) such as found in the Haskell programming language. At the other side of the spectrum, one finds arbitrary complex type specification languages (e.g. that a sorting function returns a list of type "sorted list") for which more or less powerful proof automation tools (generally first-order ones) exist.

3.3.2. Libraries

Developing libraries for programming languages takes time and generally benefits of a critical mass effect. An advantage is given to languages that start from well-established existing frameworks for which a large panel of libraries exist. Coq is such a framework.

3.4. Around and beyond the Curry-Howard correspondence

For two decades, the Curry-Howard correspondence was limited to the intuitionistic case but in 1990, an important stimulus spurred on the community following the discovery by Griffin that the correspondence was extensible to classical logic. The community then started to investigate unexplored potential fields of connection between computer science and logic. One of these fields was the computational understanding of Gentzen's sequent calculus while another one was the computational content of the axiom of choice.

3.4.1. Control operators and classical logic

Indeed, a significant extension of the Curry-Howard correspondence has been obtained at the beginning of the 90's thanks to the seminal observation by Griffin [51] that some operators known as control operators were typable by the principle of double negation elimination ($\neg\neg A \Rightarrow A$), a principle which provides classical logic.

Control operators are operators used to jump from one place of a program to another place. They were first considered in the 60's by Landin [61] and Reynolds [67] and started to be studied in an abstract way in the 80's by Felleisen *et al* [45], culminating in Parigot's $\lambda\mu$-calculus [64], a reference calculus that is in fine Curry-Howard correspondence with classical natural deduction. In this respect, control operators are fundamental pieces of the full connection between proofs and programs.

3.4.2. Sequent calculus

The Curry-Howard interpretation of sequent calculus started to be investigated at the beginning of the 90's. The main technicality of sequent calculus is the presence of *left introduction* inference rules and two kinds of interpretations of these rules are applicable. The first approach interprets left introduction rules as construction rules for a language of patterns but it does not really address the problem of the interpretation of the implication connective. The second approach, started in 1994, interprets left introduction rules as evaluation context formation rule. This line of work culminated in 2000 with the design by Hugo Herbelin and Pierre-Louis Curien of a symmetric calculus exhibiting deep dualities between the notion of programs and evaluation contexts and between the standard notions of call-by-name and call-by-value evaluation semantics.

3.4.3. Abstract machines

Abstract machines came as an intermediate evaluation device, between high-level programming languages and the computer microprocessor. The typical reference for call-by-value evaluation of λ-calculus is Landin's SECD machine [60] and Krivine's abstract machine for call-by-name evaluation [59], [57]. A typical abstract machine manipulates a state that consists of a program in some environment of bindings and some evaluation context traditionally encoded into a "stack".

3.4.4. Delimited control

Delimited control extends the expressiveness of control operators with effects: the fundamental result here is a completeness result by Filinski [46]: any side-effect expressible in monadic style (and this covers references, exceptions, states, dynamic bindings, ...) can be simulated in λ-calculus equipped with delimited control.

POLSYS Project-Team

3. Scientific Foundations

3.1. Introduction

Polynomial system solving is a fundamental problem in Computer Algebra with many applications in cryptography, robotics, biology, error correcting codes, signal theory, ... Among all available methods for solving polynomial systems, computation of Gröbner bases remains one of the most powerful and versatile method since it can be applied in the continuous case (rational coefficients) as well as in the discrete case (finite fields). Gröbner bases is also a building blocks for higher level algorithms who compute real sample points in the solution set of polynomial systems, decide connectivity queries and quantifier elimination over the reals. The major challenge facing the designer or the user of such algorithms is the intrinsic exponential behaviour of the complexity for computing Gröbner bases. The current proposal is an attempt to tackle these issues in a number of different ways: improve the efficiency of the fundamental algorithms (even when the complexity is exponential), develop high performance implementation exploiting parallel computers, and investigate new classes of structured algebraic problems where the complexity drops to polynomial time.

3.2. Fundamental Algorithms and Structured Systems

Participants: Jean-Charles Faugère, Mohab Safey El Din, Elias Tsigaridas, Guénaël Renault, Dongming Wang, Jérémy Berthomieu, Pierre-Jean Spaenlehauer, Chenqi Mou, Jules Svartz, Louise Huot, Thibault Verron.

Efficient algorithms F_4/F_5 [1] for computing the Gröbner basis of a polynomial system rely heavily on a connection with linear algebra. Indeed, these algorithms reduce the Gröbner basis computation to a sequence of Gaussian eliminations on several submatrices of the so-called Macaulay matrix in some degree. Thus, we expect to improve the existing algorithms by
(i) developing dedicated linear algebra routines performing the Gaussian elimination steps: this is precisely the objective 2 described below;
(ii) generating smaller or simpler matrices to which we will apply Gaussian elimination.
We describe here our goals for the latter problem. First, we focus on algorithms for computing a Gröbner basis of *general polynomial systems*. Next, we present our goals on the development of dedicated algorithms for computing Gröbner bases of *structured polynomial systems* which arise in various applications.

Algorithms for general systems. Several degrees of freedom are available to the designer of a Gröbner basis algorithm to generate the matrices occurring during the computation. For instance, it would be desirable to obtain matrices which would be almost triangular or very sparse. Such a goal can be achieved by considering various interpretations of the F_5 algorithm with respect to different monomial orderings. To address this problem, the tight complexity results obtained for F_5 will be used to help in the design of such a general algorithm. To illustrate this point, consider the important problem of solving boolean polynomial systems; it might be interesting to preserve the sparsity of the original equations and, at the same time, using the fact that overdetermined systems are much easier to solve.

Algorithms dedicated to *structured* polynomial systems. A complementary approach is to exploit the structure of the input polynomials to design specific algorithms. Very often, problems coming from applications are not random but are highly structured. The specific nature of these systems may vary a lot: some polynomial systems can be sparse (when the number of terms in each equation is low), overdetermined (the number of the equations is larger than the number of variables), invariants by the action of some finite groups, multi-linear (each equation is linear w.r.t. to one block of variables) or more generally multihomogeneous. In each case, the ultimate goal is to identify large classes of problems whose theoretical/practical complexity drops and to propose in each case dedicated algorithms.

[1] J.-C. Faugère. *A new efficient algorithm for computing Gröbner bases without reduction to zero (F5).* In Proceedings of ISSAC '02, pages 75-83, New York, NY, USA, 2002. ACM.

3.3. Solving Systems over the Reals and Applications.

Participants: Mohab Safey El Din, Daniel Lazard, Elias Tsigaridas, Pierre-Jean Spaenlehauer, Aurélien Greuet, Simone Naldi.

We will develop algorithms for solving polynomial systems over complex/real numbers. Again, the goal is to extend significantly the range of reachable applications using algebraic techniques based on Gröbner bases and dedicated linear algebra routines. Targeted application domains are global optimization problems, stability of dynamical systems (e.g. arising in biology or in control theory) and theorem proving in computational geometry.

The following functionalities shall be requested by the end-users:
(i) deciding the emptiness of the real solution set of systems of polynomial equations and inequalities,
(ii) quantifier elimination over the reals or complex numbers,
(iii) answering connectivity queries for such real solution sets.
We will focus on these functionalities.

We will develop algorithms based on the so-called critical point method to tackle systems of equations and inequalities (problem *(i)*) . These techniques are based on solving 0-dimensional polynomial systems encoding "critical points" which are defined by the vanishing of minors of jacobian matrices (with polynomial entries). Since these systems are highly structured, the expected results of Objective 1 and 2 may allow us to obtain dramatic improvements in the computation of Gröbner bases of such polynomial systems. This will be the foundation of practically fast implementations (based on singly exponential algorithms) outperforming the current ones based on the historical Cylindrical Algebraic Decomposition (CAD) algorithm (whose complexity is doubly exponential in the number of variables). We will also develop algorithms and implementations that allow us to analyze, at least locally, the topology of solution sets in some specific situations. A long-term goal is obviously to obtain an analysis of the global topology.

3.4. Low level implementation and Dedicated Algebraic Computation and Linear Algebra.

Participants: Jean-Charles Faugère, Christian Eder, Elias Tsigaridas, F. Martani.

Here, the primary objective is to focus on *dedicated* algorithms and software for the linear algebra steps in Gröbner bases computations and for problems arising in Number Theory. As explained above, linear algebra is a key step in the process of computing efficiently Gröbner bases. It is then natural to develop specific linear algebra algorithms and implementations to further strengthen the existing software. Conversely, Gröbner bases computation is often a key ingredient in higher level algorithms from Algebraic Number Theory. In theses cases, the algebraic problems are very particular and specific. Hence dedicated Gröbner bases algorithms and implementations would provide a better efficiency.

Dedicated linear algebra tools. FGB is an efficient library for Gröbner bases computations which can be used, for instance, via MAPLE. However, the library is sequential. A goal of the project is to extend its efficiency to new trend parallel architectures such as clusters of multi-processor systems in order to tackle a broader class of problems for several applications. Consequently, our first aim is to provide a durable, long term software solution, which will be the successor of the existing FGB library. To achieve this goal, we will first develop a high performance linear algebra package (under the LGPL license). This could be organized in the form of a collaborative project between the members of the team. The objective is not to develop a general library similar to the LINBOX project but to propose a dedicated linear algebra package taking into account the specific properties of the matrices generated by the Gröbner bases algorithms. Indeed these matrices are sparse (the actual sparsity depends strongly on the application), almost block triangular and not necessarily of full rank. Moreover, most of the pivots are known at the beginning of the computation. In practice, such matrices are huge (more than 10^6 columns) but taking into account their shape may allow us to speed up the computations by one or several orders of magnitude. A variant of a Gaussian elimination algorithm together with a corresponding C implementation has been presented. The main peculiarity is the order in which the operations are performed. This will be the kernel of the new linear library that will be developed.

Fast linear algebra packages would also benefit to the transformation of a Gröbner basis of a zero–dimensional ideal with respect to a given monomial ordering into a Gröbner basis with respect to another ordering. In the generic case at least, the change of ordering is equivalent to the computation of the minimal polynomial of a so-called multiplication matrix. By taking into account the sparsity of this matrix, the computation of the Gröbner basis can be done more efficiently using variant of the Wiedemann algorithm. Hence, our goal is also to obtain a dedicated high performance library for transforming (i.e. change ordering) Gröbner bases.

Dedicated algebraic tools for Algebraic Number Theory. Recent results in Algebraic Number Theory tend to show that the computation of Gröbner bases is a key step toward the resolution of difficult problems in this domain [2]. Using existing resolution methods is simply not enough to solve relevant problems. The main algorithmic lock to overcome is to adapt the Gröbner basis computation step to the specific problems. Typically, problems coming from Algebraic Number Theory usually have a lot of symmetries or the input systems are very structured. This is the case in particular for problems coming from the algorithmic theory of Abelian varieties over finite fields [3] where the objects are represented by polynomial system and are endowed with intrinsic group actions. The main goal here is to provide dedicated algebraic resolution algorithms and implementations for solving such problems. We do not restrict our focus on problems in positive characteristic. For instance, tower of algebraic fields can be viewed as triangular sets; more generally, related problems (e.g. effective Galois theory) which can be represented by polynomial systems will receive our attention. This is motivated by the fact that, for example, computing small integer solutions of Diophantine polynomial systems in connection with Coppersmith's method would also gain in efficiency by using a dedicated Gröbner bases computations step.

3.5. Solving Systems in Finite Fields, Applications in Cryptology and Algebraic Number Theory.

Participants: Jean-Charles Faugère, Ludovic Perret, Guénaël Renault, Louise Huot, Frédéric de Portzamparc, Rina Zeitoun.

Here, we focus on solving polynomial systems over finite fields (i.e. the discrete case) and the corresponding applications (Cryptology, Error Correcting Codes, ...). Obviously this objective can be seen as an application of the results of the two previous objectives. However, we would like to emphasize that it is also the source of new theoretical problems and practical challenges. We propose to develop a systematic use of *structured systems* in *algebraic cryptanalysis*.
(i) So far, breaking a cryptosystem using algebraic techniques could be summarized as modeling the problem by algebraic equations and then computing a, usually, time consuming Gröbner basis. A new trend in this field is to require a theoretical complexity analysis. This is needed to explain the behavior of the attack but also to help the designers of new cryptosystems to propose actual secure parameters.
(ii) To assess the security of several cryptosystems in symmetric cryptography (block ciphers, hash functions, ...), a major difficulty is the size of the systems involved for this type of attack. More specifically, the bottleneck is the size of the linear algebra problems generated during a Gröbner basis computation.

We propose to develop a systematic use of *structured systems* in *algebraic cryptanalysis*.

[2] P. Gaudry, *Index calculus for abelian varieties of small dimension and the elliptic curve discrete logarithm problem*, Journal of Symbolic Computation 44,12 (2009) pp. 1690-1702
[3] e.g. point counting, discrete logarithm, isogeny.

The first objective is to build on the recent breakthrough in attacking McEliece's cryptosystem: it is the first structural weakness observed on one of the oldest public key cryptosystem. We plan to develop a well founded framework for assessing the security of public key cryptosystems based on coding theory from the algebraic cryptanalysis point of view. The answer to this issue is strongly related to the complexity of solving bihomogeneous systems (of bidegree $(1, d)$). We also plan to use the recently gained understanding on the complexity of structured systems in other areas of cryptography. For instance, the MinRank problem – which can be modeled as an overdetermined system of bilinear equations – is at the heart of the structural attack proposed by Kipnis and Shamir against HFE (one of the most well known multivariate public cryptosystem). The same family of structured systems arises in the algebraic cryptanalysis of the Discrete Logarithmic Problem (DLP) over curves (defined over some finite fields). More precisely, some bilinear systems appear in the polynomial modeling the points decomposition problem. Moreover, in this context, a natural group action can also be used during the resolution of the considered polynomial system.

Dedicated tools for linear algebra problems generated during the Gröbner basis computation will be used in algebraic cryptanalysis. The promise of considerable algebraic computing power beyond the capability of any standard computer algebra system will enable us to attack various cryptosystems or at least to propose accurate secure parameters for several important cryptosystems. Dedicated linear tools are thus needed to tackle these problems. From a theoretical perspective, we plan to further improve the theoretical complexity of the hybrid method and to investigate the problem of solving polynomial systems with noise, i.e. some equations of the system are incorrect. The hybrid method is a specific method for solving polynomial systems over finite fields. The idea is to mix exhaustive search and Gröbner basis computation to take advantage of the over-determinacy of the resulting systems.

Polynomial system with noise is currently emerging as a problem of major interest in cryptography. This problem is a key to further develop new applications of algebraic techniques; typically in side-channel and statistical attacks. We also emphasize that recently a connection has been established between several classical lattice problems (such as the Shortest Vector Problem), polynomial system solving and polynomial systems with noise. The main issue is that there is no sound algorithmic and theoretical framework for solving polynomial systems with noise. The development of such framework is a long-term objective.

PROSECCO Project-Team

3. Scientific Foundations

3.1. Symbolic verification of cryptographic applications

Despite decades of experience, designing and implementing cryptographic applications remains dangerously error-prone, even for experts. This is partly because cryptographic security is an inherently hard problem, and partly because automated verification tools require carefully-crafted inputs and are not widely applicable. To take just the example of TLS, a widely-deployed and well-studied cryptographic protocol designed, implemented, and verified by security experts, the lack of a formal proof about all its details has regularly led to the discovery of major attacks (in 2003, 2008, 2009, and 2011) on both the protocol and its implementations, after many years of unsuspecting use.

As a result, the automated verification for cryptographic applications is an active area of research, with a wide variety of tools being employed for verifying different kinds of applications.

In previous work, the we have developed the following three approaches:

- ProVerif: a symbolic prover for cryptographic protocol models
- Tookan: an attack-finder for PKCS#11 hardware security devices
- F7: a security typechecker for cryptographic applications written in F#

3.1.1. Verifying cryptographic protocols with ProVerif

Given a model of a cryptographic protocol, the problem is to verify that an active attacker, possibly with access to some cryptographic keys but unable to guess other secrets, cannot thwart security goals such as authentication and secrecy [52]; it has motivated a serious research effort on the formal analysis of cryptographic protocols, starting with [50] and eventually leading to effective verification tools, such as our tool ProVerif.

To use ProVerif, one encodes a protocol model in a formal language, called the applied pi-calculus, and ProVerif abstracts it to a set of generalized Horn clauses. This abstraction is a small approximation: it just ignores the number of repetitions of each action, so ProVerif is still very precise, more precise than, say, tree automata-based techniques. The price to pay for this precision is that ProVerif does not always terminate; however, it terminates in most cases in practice, and it always terminates on the interesting class of *tagged protocols* [46]. ProVerif also distinguishes itself from other tools by the variety of cryptographic primitives it can handle, defined by rewrite rules or by some equations, and the variety of security properties it can prove: secrecy [44], [38], correspondences (including authentication) [45], and observational equivalences [43]. Observational equivalence means that an adversary cannot distinguish two processes (protocols); equivalences can be used to formalize a wide range of properties, but they are particularly difficult to prove. Even if the class of equivalences that ProVerif can prove is limited to equivalences between processes that differ only by the terms they contain, these equivalences are useful in practice and ProVerif is the only tool that proves equivalences for an unbounded number of sessions.

Using ProVerif, it is now possible to verify large parts of industrial-strength protocols such as TLS [13], JFK [39], and Web Services Security [42]. against powerful adversaries that can run an unlimited number of protocol sessions, for strong security properties expressed as correspondence queries or equivalence assertions. ProVerif is used by many teams at the international level, and has been used in more 30 research papers (references available at http://proverif.inria.fr/proverif-users.html).

3.1.2. Verifying security APIs using Tookan

Security application programming interfaces (APIs) are interfaces that provide access to functionality while also enforcing a security policy, so that even if a malicious program makes calls to the interface, certain security properties will continue to hold. They are used, for example, by cryptographic devices such as smartcards and Hardware Security Modules (HSMs) to manage keys and provide access to cryptographic functions whilst keeping the keys secure. Like security protocols, their design is security critical and very difficult to get right. Hence formal techniques have been adapted from security protocols to security APIs.

The most widely used standard for cryptographic APIs is RSA PKCS#11, ubiquitous in devices from smartcards to HSMs. A 2003 paper highlighted possible flaws in PKCS#11 [48], results which were extended by formal analysis work using a Dolev-Yao style model of the standard [49]. However at this point it was not clear to what extent these flaws affected real commercial devices, since the standard is underspecified and can be implemented in many different ways. The Tookan tool, developed by Steel in collaboration with Bortolozzo, Centenaro and Focardi, was designed to address this problem. Tookan can reverse engineer the particular configuration of PKCS#11 used by a device under test by sending a carefully designed series of PKCS#11 commands and observing the return codes. These codes are used to instantiate a Dolev-Yao model of the device's API. This model can then be searched using a security protocol model checking tool to find attacks. If an attack is found, Tookan converts the trace from the model checker into the sequence of PKCS#11 queries needed to make the attack and executes the commands directly on the device. Results obtained by Tookan are remarkable: of 18 commercially available PKCS#11 devices tested, 10 were found to be susceptible to at least one attack.

3.1.3. Verifying cryptographic applications using F7

Verifying the implementation of a protocol has traditionally been considered much harder than verifying its model. This is mainly because implementations have to consider real-world details of the protocol, such as message formats, that models typically ignore. This leads to a situation that a protocol may have been proved secure in theory, but its implementation may be buggy and insecure. However, with recent advances in both program verification and symbolic protocol verification tools, it has become possible to verify fully functional protocol implementations in the symbolic model.

One approach is to extract a symbolic protocol model from an implementation and then verify the model, say, using ProVerif. This approach has been quite successful, yielding a verified implementation of TLS in F# [13]. However, the generated models are typically quite large and whole-program symbolic verification does not scale very well.

An alternate approach is to develop a verification method directly for implementation code, using well-known program verification techniques such as typechecking. F7 [40] is a refinement typechecker for F#, developed jointly at Microsoft Research Cambridge and Inria. It implements a dependent type-system that allows us to specify security assumptions and goals as first-order logic annotations directly inside the program. It has been used for the modular verification of large web services security protocol implementations [41]. F* [53] is an extension of F7 with higher-order kinds and a certifying typechecker. Both F7 and F* have a growing user community. The cryptographic protocol implementations verified using F7 and F* already represent the largest verified cryptographic applications to our knowledge.

3.2. Computational verification of cryptographic applications

Proofs done by cryptographers in the computational model are mostly manual. Our goal is to provide computer support to build or verify these proofs. In order to reach this goal, we have already designed the automatic tool CryptoVerif, which generates proofs by sequences of games. Much work is still needed in order to develop this approach, so that it is applicable to more protocols. We also plan to design and implement techniques for proving implementations of protocols secure in the computational model, by generating them from CryptoVerif specifications that have been proved secure, or by automatically extracting CryptoVerif models from implementations.

An alternative approach is to directly verify cryptographic applications in the computational model by typing. A recent work [51] shows how to use refinement typechecking in F7 to prove computational security for protocol implementations. In this method, henceforth referred to as computational F7, typechecking is used as the main step to justify a classic game-hopping proof of computational security. The correctness of this method is based on a probabilistic semantics of F# programs and crucially relies on uses of type abstraction and parametricity to establish strong security properties, such as indistinguishability.

In principle, the two approaches, typechecking and game-based proofs, are complementary. Understanding how to combine these approaches remains an open and active topic of research.

3.3. Provably secure web applications

Web applications are fast becoming the dominant programming platform for new software, probably because they offer a quick and easy way for developers to deploy and sell their *app*s to a large number of customers. Third-party web-based apps for Facebook, Apple, and Google, already number in the hundreds of thousands and are likely to grow in number. Many of these applications store and manage private user data, such as health information, credit card data, and GPS locations. To protect this data, applications tend to use an ad hoc combination of cryptographic primitives and protocols. Since designing cryptographic applications is easy to get wrong even for experts, we believe this is an opportune moment to develop security libraries and verification techniques to help web application programmers.

As a typical example, consider commercial password managers, such as LastPass, RoboForm, and 1Password. They are implemented as browser-based web applications that, for a monthly fee, offer to store a user's passwords securely on the web and synchronize them across all of the user's computers and smartphones. The passwords are encrypted using a master password (known only to the user) and stored in the cloud. Hence, no-one except the user should ever be able to read her passwords. When the user visits a web page that has a login form, the password manager asks the user to decrypt her password for this website and automatically fills in the login form. Hence, the user no longer has to remember passwords (except her master password) and all her passwords are available on every computer she uses.

Password managers are available as browser extensions for mainstream browsers such as Firefox, Chrome, and Internet Explorer, and as downloadable apps for Android and Apple phones. So, seen as a distributed application, each password manager application consists of a web service (written in PHP or Java), some number of browser extensions (written in JavaScript), and some smartphone apps (written in Java or Objective C). Each of these components uses a different cryptographic library to encrypt and decrypt password data. How do we verify the correctness of all these components?

We propose three approaches. For client-side web applications and browser extensions written in JavaScript, we propose to build a static and dynamic program analysis framework to verify security invariants. For Android smartphone apps and web services written in Java, we propose to develop annotated JML cryptography libraries that can be used with static analysis tools like ESC/Java to verify the security of application code. For clients and web services written in F# for the .NET platform, we propose to use F7 to verify their correctness.

SECRET Project-Team

3. Scientific Foundations

3.1. Scientific foundations

Our research work is mainly devoted to the design and analysis of cryptographic algorithms. Our approach on the previous problems relies on a competence whose impact is much wider than cryptology. Our tools come from information theory, discrete mathematics, probabilities, algorithmics... Most of our work mix fundamental aspects (study of mathematical objects) and practical aspects (cryptanalysis, design of algorithms, implementations). Our research is mainly driven by the belief that discrete mathematics and algorithmics of finite structures form the scientific core of (algorithmic) data protection.

CAD Team

3. Scientific Foundations

3.1. Geometry continuity and ε Geometry Continuity

The mathematical background of parametric surfaces is Differential Geometry. In differential geometry, Riemann (1826 1866), Shiing-Shen Chern (1911 2004), continuities play a very important kernel role. In 1980s, more and more engineering design using geometry modeling softwares found the problems of the parametric continuities. And the order of the parametric continuity depends on how the curve is parameterized. To day, engineers and scientists try to find a kind of continuities, which are the intuitive intrinsic properties of curves and surfaces, and the orders of the continuities are independent of the parameterization.

G-Continuity could be defined as the smoothness properties of a curve or a surface that are more than its order of differentiability. This problem is complex and progress in this domain is very slow. We proposed new ways to make through the bottleneck. Furthermore, we also wanted to fill the gap between the traditional mathematics and modern computer science. Hence, we developed the theories of epsilon-geometry continuities to accommodate the representation and the rounding errors of float-point arithmetic, and design new geometric modeling operators under the constraints of epsilon-geometry continuities.

3.2. Two main challenges in Computer Aided Design

3.2.1. Robustness tolerance, error control

Based on this theoretical contribution, we also proposed several elegant solutions to the most important challenges in Computer Aided Design (see Lees A Piegl. "Ten challenges in Computer-Aided-Design". Jal of CAD 2005. 37 (4): 461-470): robustness, tolerances, error control, geometric arrangement, beautification and modelling of complex shapes. During CAD processes one uses a myriad of tolerances, many of which are not directly related to the actual manufacturing process. Some interesting questions here include: What are the most relevant machining tolerances? How to set the army of computational tolerances, e.g. those of systems of equations, to guarantee machining within the required accuracy? How tolerances in different spaces, e.g. in model space and in parameter space, are related. Numerical instabilities also account for the majority of computational errors in commercial CAD systems.

The problems related to robustness haunt every programmer who has ever worked on commercial systems. Fixing numerical bugs can be very frustrating, and often times results in patching up the code simply because no solution exists to remedy the problem.

3.2.2. Geometry beautification, Geometry operators and Shape generation

Although geometric uncertainties are related to robustness and tolerance, there are a number of extra issues well worth deeper investigations. Geometric arrangements are full of special cases. The most notable ones are: cases of touch, overlapping, containment, etc.; cases of parallelism, perpendicularity, coincidence, etc.; axes of symmetrical data, data clustering, dense or sparse data, etc.; cases of degeneracy, discontinuity, inconsistencies, etc.; problems with cracks, excess material, lack of detail, etc. In just about any code that deals with geometry, the number of special cases is significantly larger than the general ones. Data explosion is the result of careless selection of the methods, e.g. parameter space-based sampling, and improper implementation, e.g. recursive algorithms. Some of the relevant issues are: sampling: over sampling, sampling in incorrect places, etc; procedural definitions, e.g. lofting a large set of curves or merging surfaces may result in an explosion of control points.

Furthermore, although CAD processes are supposed to produce valid and "made to order" models, the reality is that most (if not all) models are rough and require post-processing, i.e. beautification. Some of the most frequently needed tasks are: removing unwanted edges, corners, cracks, etc.; removing bumps, oscillations, curvature extremes, etc.; healing incorrect models, e.g. removing holes in triangulations; smoothing, fairing, re-shaping, etc.

3.3. Computer Graphics

In Computer Graphics, objectives were to prove the capability of the team to address some topics as Computational Photography, Rendering and Computer Animation. Work in Progress in these topics are described in the following chapter.

CLASSIC Project-Team

3. Scientific Foundations

3.1. Regression models of supervised learning

The most obvious contribution of statistics to machine learning is to consider the supervised learning scenario as a special case of regression estimation: given n independent pairs of observations (X_i, Y_i), $i = 1, \cdots, n$, the aim is to "learn" the dependence of Y_i on X_i. Thus, classical results about statistical regression estimation apply, with the caveat that the hypotheses we can reasonably assume about the distribution of the pairs (X_i, Y_i) are much weaker than what is usually considered in statistical studies. The aim here is to assume very little, maybe only independence of the observed sequence of input-output pairs, and to validate model and variable selection schemes. These schemes should produce the best possible approximation of the joint distribution of (X_i, Y_i) within some restricted family of models. Their performance is evaluated according to some measure of discrepancy between distributions, a standard choice being to use the Kullback-Leibler divergence.

3.1.1. PAC-Bayes inequalities

One of the specialties of the team in this direction is to use PAC-Bayes inequalities to combine thresholded exponential moment inequalities. The name of this theory comes from its founder, David McAllester, and may be misleading. Indeed, its cornerstone is rather made of non-asymptotic entropy inequalities, and a perturbative approach to parameter estimation. The team has made major contributions to the theory, first focussed on classification [6], then on regression [1]. It has introduced the idea of combining the PAC-Bayesian approach with the use of thresholded exponential moments, in order to derive bounds under very weak assumptions on the noise.

3.1.2. Sparsity and ℓ_1–regularization

Another line of research in regression estimation is the use of sparse models, and its link with ℓ_1–regularization. Regularization is the joint minimization of some empirical criterion and some penalty function; it should lead to a model that not only fits well the data but is also as simple as possible.

For instance, the Lasso uses a ℓ^1–regularization instead of a ℓ^0–one; it is popular mostly because it leads to *sparse* solutions (the estimate has only a few nonzero coordinates), which usually have a clear interpretation in many settings (e.g., the influence or lack of influence of some variables). In addition, unlike ℓ^0–penalization, the Lasso is *computationally feasible* for high-dimensional data.

3.1.3. Pushing it to the extreme: no assumption on the data

The next brick of our scientific foundations explains why and how, in certains cases, we may formulate absolutely no assumption on the data (x_i, y_i), $i = 1, \cdots, n$, which is then considered a deterministic set of input–output pairs.

3.2. On-line aggregation of predictors for the prediction of time series, with or without stationarity assumptions

We are concerned here with *sequential prediction* of outcomes, given some base predictions formed by *experts*. We distinguish two settings, depending on how the sequence of outcomes is generated: it is either

- the realization of some stationary process,
- or is not modeled at all as the realization of any underlying stochastic process (these sequences are called *individual sequences*).

The aim is to predict almost as well as the best expert. Typical good forecasters maintain one weight per expert, update these weights depending on the past performances, and output at each step the corresponding weighted linear combination of experts' advices.

The difference between the cumulative prediction error of the forecaster and the one of the best expert is called the regret. The goal here is to upper bound the regret by a quantity as small as possible.

3.3. Multi-armed bandit problems, prediction with limited feedback

We are interested in settings in which the feedback obtained on the predictions is limited, in the sense that it does not fully reveal what actually happened.

3.3.1. Bandit problems

This is also a sequential problem in which some regret is to be minimized.

However, this problem is a stochastic problem: a large number of arms, possibly indexed by a continuous set like $[0, 1]$, is available. Each arm is associated with a fixed but unknown distribution. At each round, the player chooses an arm, a payoff is drawn at random according to the distribution that is associated with it, and the only feedback that the player gets is the value of this payoff. The key quantity to study this problem is the mean-payoff function f, that indicates for each arm x the expected payoff $f(x)$ of the distribution that is associated with it. The target is to minimize the regret, i.e., ensure that the difference between the cumulative payoff obtained by the player and the one of the best arm is small.

3.3.2. A generalization of the regret: the approachability of sets

Approachability is the ability to control random walks. At each round, a vector payoff is obtained by the first player, depending on his action and on the action of the opponent player. The aim is to ensure that the average of the vector payoffs converges to some convex set. Necessary and sufficient conditions were obtained by Blackwell and others to ensure that such strategies exist, both in the full information and in the bandit cases.

Some of these results can be extended to the case of games with signals (games with partial monitoring), where at each round the only feedback obtained by the first player is a random signal drawn according to a distribution that depends on the action profile taken by the two players, while the opponent player still has a full monitoring.

GAMMA3 Project-Team (section vide)

MATHRISK Team (section vide)

MICMAC Project-Team

3. Scientific Foundations

3.1. Scientific Foundations

Quantum Chemistry aims at understanding the properties of matter through the modeling of its behavior at a subatomic scale, where matter is described as an assembly of nuclei and electrons. At this scale, the equation that rules the interactions between these constitutive elements is the Schrödinger equation. It can be considered (except in few special cases notably those involving relativistic phenomena or nuclear reactions) as a universal model for at least three reasons. First it contains all the physical information of the system under consideration so that any of the properties of this system can in theory be deduced from the Schrödinger equation associated to it. Second, the Schrödinger equation does not involve any empirical parameters, except some fundamental constants of Physics (the Planck constant, the mass and charge of the electron, ...); it can thus be written for any kind of molecular system provided its chemical composition, in terms of natures of nuclei and number of electrons, is known. Third, this model enjoys remarkable predictive capabilities, as confirmed by comparisons with a large amount of experimental data of various types. On the other hand, using this high quality model requires working with space and time scales which are both very tiny: the typical size of the electronic cloud of an isolated atom is the Angström (10^{-10} meters), and the size of the nucleus embedded in it is 10^{-15} meters; the typical vibration period of a molecular bond is the femtosecond (10^{-15} seconds), and the characteristic relaxation time for an electron is 10^{-18} seconds. Consequently, Quantum Chemistry calculations concern very short time (say 10^{-12} seconds) behaviors of very small size (say 10^{-27} m^3) systems. The underlying question is therefore whether information on phenomena at these scales is useful in understanding or, better, predicting macroscopic properties of matter. It is certainly not true that *all* macroscopic properties can be simply upscaled from the consideration of the short time behavior of a tiny sample of matter. Many of them derive from ensemble or bulk effects, that are far from being easy to understand and to model. Striking examples are found in solid state materials or biological systems. Cleavage, the ability minerals have to naturally split along crystal surfaces (e.g. mica yields to thin flakes) is an ensemble effect. Protein folding is also an ensemble effect that originates from the presence of the surrounding medium; it is responsible for peculiar properties (e.g. unexpected acidity of some reactive site enhanced by special interactions) upon which vital processes are based. However, it is undoubtedly true that *many* macroscopic phenomena originate from elementary processes which take place at the atomic scale. Let us mention for instance the fact that the elastic constants of a perfect crystal or the color of a chemical compound (which is related to the wavelengths absorbed or emitted during optic transitions between electronic levels) can be evaluated by atomic scale calculations. In the same fashion, the lubricative properties of graphite are essentially due to a phenomenon which can be entirely modeled at the atomic scale. It is therefore reasonable to simulate the behavior of matter at the atomic scale in order to understand what is going on at the macroscopic one. The journey is however a long one. Starting from the basic principles of Quantum Mechanics to model the matter at the subatomic scale, one finally uses statistical mechanics to reach the macroscopic scale. It is often necessary to rely on intermediate steps to deal with phenomena which take place on various *mesoscales*. It may then be possible to couple one description of the system with some others within the so-called *multiscale* models. The sequel indicates how this journey can be completed focusing on the first smallest scales (the subatomic one), rather than on the larger ones. It has already been mentioned that at the subatomic scale, the behavior of nuclei and electrons is governed by the Schrödinger equation, either in its time dependent form or in its time independent form. Let us only mention at this point that

- both equations involve the quantum Hamiltonian of the molecular system under consideration; from a mathematical viewpoint, it is a self-adjoint operator on some Hilbert space; *both* the Hilbert space and the Hamiltonian operator depend on the nature of the system;
- also present into these equations is the wavefunction of the system; it completely describes its state; its L^2 norm is set to one.

The time dependent equation is a first order linear evolution equation, whereas the time-independent equation is a linear eigenvalue equation. For the reader more familiar with numerical analysis than with quantum mechanics, the linear nature of the problems stated above may look auspicious. What makes the numerical simulation of these equations extremely difficult is essentially the huge size of the Hilbert space: indeed, this space is roughly some symmetry-constrained subspace of $L^2(\mathbb{R}^d)$, with $d = 3(M + N)$, M and N respectively denoting the number of nuclei and the number of electrons the system is made of. The parameter d is already 39 for a single water molecule and rapidly reaches 10^6 for polymers or biological molecules. In addition, a consequence of the universality of the model is that one has to deal at the same time with several energy scales. In molecular systems, the basic elementary interaction between nuclei and electrons (the two-body Coulomb interaction) appears in various complex physical and chemical phenomena whose characteristic energies cover several orders of magnitude: the binding energy of core electrons in heavy atoms is 10^4 times as large as a typical covalent bond energy, which is itself around 20 times as large as the energy of a hydrogen bond. High precision or at least controlled error cancellations are thus required to reach chemical accuracy when starting from the Schrödinger equation. Clever approximations of the Schrödinger problems are therefore needed. The main two approximation strategies, namely the Born-Oppenheimer-Hartree-Fock and the Born-Oppenheimer-Kohn-Sham strategies, end up with large systems of coupled *nonlinear* partial differential equations, each of these equations being posed on $L^2(\mathbb{R}^3)$. The size of the underlying functional space is thus reduced at the cost of a dramatic increase of the mathematical complexity of the problem: nonlinearity. The mathematical and numerical analysis of the resulting models has been the major concern of the project-team for a long time. In the recent years, while part of the activity still follows this path, the focus has progressively shifted to problems at other scales. Such problems are described in the following sections.

SIERRA Project-Team

3. Scientific Foundations

3.1. Supervised Learning

This part of our research focuses on methods where, given a set of examples of input/output pairs, the goal is to predict the output for a new input, with research on kernel methods, calibration methods, and multi-task learning.

3.2. Unsupervised Learning

We focus here on methods where no output is given and the goal is to find structure of certain known types (e.g., discrete or low-dimensional) in the data, with a focus on matrix factorization, statistical tests, dimension reduction, and semi-supervised learning.

3.3. Parsimony

The concept of parsimony is central to many areas of science. In the context of statistical machine learning, this takes the form of variable or feature selection. The team focuses primarily on structured sparsity, with theoretical and algorithmic contributions (this is the main topic of the ERC starting investigator grant awarded to F. Bach).

3.4. Optimization

Optimization in all its forms is central to machine learning, as many of its theoretical frameworks are based at least in part on empirical risk minimization. The team focuses primarily on convex and bandit optimization, with a particular focus on large-scale optimization.

BANG Project-Team

3. Scientific Foundations

3.1. Introduction

The dynamics of complex physical or biophysical phenomena involving many particles, including biological cells - which can be seen as active particles -, can be represented efficiently either by explicitly considering the behaviour of each particle individually or by Partial Differential Equations which, under certain hypotheses, represent local averages over a sufficiently large number of particles.

Since the XIXth century this formalism has shown its efficiency and ability to explain both qualitative and quantitative behaviours. The knowledge that has been gathered on such physical models, on algorithms for solving them on computers, on industrial implementation, opens the hope for success when dealing with life sciences also. This is one of the main goals of BANG. At small spatial scales, or at spatial scales of individual matter components where heterogeneities in the medium occur, agent-based models are developed. They complement the partial differential equation models considered on scales at which averages over the individual components behave sufficiently smoothly.

3.2. Mathematical modelling

What are the relevant physical or biological variables, what are the possible dominant effects ruling their dynamics, how to analyse the information coming out from a mathematical model and interpret them in the real situations under consideration ? These are the questions leading to select a mathematical model, generally also to couple several of them in order to render all physical or biomedical features which are selected by specialist partners (engineers, physicists, medical doctors). These are usually based on Navier-Stokes system for fluids (as in free surface fluid flows), on parabolic-hyperbolic equations (Saint-Venant system for shallow water, flows of electrons/holes in semiconductors, Keller-Segel model of chemotaxis).

3.3. Multiscale analysis

The complete physical or biomedical description is usually complex and requires very small scales. Efficiency of computer resolution leads to simplifications using averages of quantities. Methods allowing to achieve that goal are numerous and mathematically deep. Some examples studied in BANG are

- Coupled multiscale modelling (description of tumours and tissues from the sub-cellular level to the organ scale).
- Description of cell movement from the individual to the collective scales.
- Reduction of full 3d Navier-Stokes system to 2d or 1d hyperbolic equations by a section average (derivation of Saint-Venant system for shallow water).

3.4. Numerical Algorithms

Various numerical methods are used in BANG. They may be based on finite elements or finite volume methods, or stochastic methods for individual agents. Algorithmic improvements are needed in order to take into account the specificity of each model, of their coupling, or their 3D features. Among them we can mention

- Well-balanced schemes for shallow water system.
- Free-surface Navier-Stokes solvers based on a multilayer St-Venant approach.
- Deterministic and stochastic agent based models for the simulation of multi-cellular systems.

3.5. Proliferation dynamics and its control

- Cell division cycle in structured cell populations.
- Physiological and pharmacological control of cell proliferation.
- Physical mechanisms and constraints in cell proliferation.
- Intracellular spatiotemporal dynamics of genes and proteins: p53.
- Cell darwinism and drug resistance in cancer cells.
- Optimisation of cancer chemotherapy.
- Protein polymerisation and application to amyloid diseases.
- Inverse Problem for growth-fragmentation equations.

3.6. Tissue growth, regeneration and cell movements

This research activity aims at studying mathematical models related to tumour development and tissue organisation. Among the many biological aspects, examples are:

- Biomedical aspects of cell-cell interactions at the local and whole organ level.
- Migration of cells in tissues.
- Growth control of living tissues and organs.
- Regenerative medicine.
- Early embryology, and biomechanical aspects of cell interaction.
- Chemotaxis, self-organisation in cell populations.

3.7. Neurosciences

Cortical networks are constituted of a large number of statistically similar neurons in interaction. Each neuron has a nonlinear dynamics and is subject to noise. Moreover, neurological treatment involve several timescales. Multiscale analysis, both in spatial (number of cells) and temporal hence also constitute mathematical foundations of our approaches to neurosciences. In addition to the techniques described in section 3.1 - 3.4, our approach of the activity of large cortical areas involve:

- limit theorems of stochastic interacting particles systems, such as coupling methods or large deviations techniques, as used in mathematical approaches to the statistical physics of gases
- bifurcation analysis of deterministic and stochastic differential equation used to analyze the qualitative behaviour of networks
- singular perturbation theory, geometrical and topological approaches in dynamical systems used to uncover the dynamics in the presence of multiple timescales.

3.8. Free surface flows

Several industrial applications require to solve fluid flows with a free surface. BANG develops algorithms in two directions. Firstly flows in rivers and coastal areas using the Saint-Venant model with applications to dam break and pollution problems in averaged shallow water systems. Secondly, 3D hydrostatic flows by a multilayer Saint-Venant approach and 3D Navier-Stokes flows.

CLIME Project-Team

3. Scientific Foundations

3.1. Data assimilation and inverse modeling

This activity is one major concern of environmental sciences. It matches up the setting and the use of data assimilation methods, for instance variational methods (such as the 4D-Var method). An emerging issue lies in the propagation of uncertainties by models, notably through ensemble forecasting methods.

Although modeling is not part of the scientific objectives of Clime, the project-team has complete access to models developed by CEREA: the models from Polyphemus (pollution forecasting from local to regional scales) and Code_Saturne (urban scale). In regard to other modeling domains, such as meteorology and oceanography, Clime accesses models through co-operation initiatives.

The research activities of Clime tackle scientific issues such as:

- Within a family of models (differing by their physical formulations and numerical approximations), which is the optimal model for a given set of observations?
- How to reduce dimensionality of problems by Galerkin projection of equations on subspaces? How to define these subspaces in order to keep the main properties of systems?
- How to assess the quality of a forecast and its uncertainty? How do data quality, missing data, data obtained from sub-optimal locations, affect the forecast? How to better include information on uncertainties (of data, of models) within the data assimilation system?
- How to make a forecast (and a better forecast!) by using several models corresponding to different physical formulations? It also raises the question: how should data be assimilated in this context?
- Which observational network should be set up to perform a better forecast, while taking into account additional criteria such as observation cost? What are the optimal location, type and mode of deployment of sensors? How should trajectories of mobile sensors be operated, while the studied phenomenon is evolving in time? This issue is usually referred as “network design”.

3.2. Satellite acquisitions and image assimilation

In geosciences, the issue of coupling data, in particular satellite acquisitions, and models is extensively studied for meteorology, oceanography, chemistry-transport and land surface models. However, satellite images are mostly assimilated on a point-wise basis. Three major approaches arise if taking into account the spatial structures, whose displacement is visualized on image sequences:

- Image approach. Image assimilation allows the extraction of features from image sequences, for instance motion field or structures’ trajectory. A model of the dynamics is considered (obtained by simplification of a geophysical model such as Navier-Stokes equations). An observation operator is defined to express the links between the model state and the pixel value. In the simplest case, the pixel value corresponds to one coordinate of the model state and the observation operator is reduced to a projection. However, in most cases, this operator is highly complex, implicit and non-linear. Data assimilation techniques are developed to control the initial state or the whole assimilation window. Image assimilation is also applied to learn reduced models from image data and estimate a reliable and small-size reconstruction of the dynamics, which is observed on the sequence.
- Model approach. Image assimilation is used to control an environmental model and obtain improved forecasts. In order to take into account the spatial and temporal coherency of structures, specific image characteristics are considered, and dedicated norms and observation error covariances are defined.

- Correcting a model. Another topic, mainly described for meteorology in the literature, concerns the location of structures. How to force the existence and to correct the location of structures in the model state using image information? Most of the operational meteorological forecasting institutes, such as MétéoFrance, UK-met, KNMI (in Netherlands), ZAMG (in Austria) and Met-No (in Norway), study this issue because operational forecasters often modify their forecasts based on visual comparisons between the model outputs and the structures displayed on satellite images.

3.3. Software chains for environmental applications

An objective of Clime is to participate in the design and creation of software chains for impact assessment and environmental crisis management. Such software chains bring together static or dynamic databases, data assimilation systems, forecast models, processing methods for environmental data and images, complex visualization tools, scientific workflows, ...

Clime is currently building, in partnership with École des Ponts ParisTech and EDF R&D, such a system for air pollution modeling: Polyphemus (see the web site http://cerea.enpc.fr/polyphemus/), whose architecture is specified to satisfy data requirements (e.g., various raw data natures and sources, data preprocessing) and to support different uses of an air quality model (e.g., forecasting, data assimilation, ensemble runs).

POMDAPI Project-Team (section vide)

REO Project-Team

3. Scientific Foundations

3.1. Multiphysics modeling

In large vessels and in large bronchi, blood and air flows are generally supposed to be governed by the incompressible Navier-Stokes equations. Indeed in large arteries, blood can be supposed to be Newtonian, and at rest air can be modeled as an incompressible fluid. The cornerstone of the simulations is therefore a Navier-Stokes solver. But other physical features have also to be taken into account in simulations of biological flows, in particular fluid-structure interaction in large vessels and transport of sprays, particles or chemical species.

3.1.1. Fluid-structure interaction

Fluid-structure coupling occurs both in the respiratory and in the circulatory systems. We focus mainly on blood flows since our work is more advanced in this field. But the methods developed for blood flows could be also applied to the respiratory system.

Here "fluid-structure interaction" means a coupling between the 3D Navier-Stokes equations and a 3D (possibly thin) structure in large displacements.

The numerical simulations of the interaction between the artery wall and the blood flows raise many issues: (1) the displacement of the wall cannot be supposed to be infinitesimal, geometrical nonlinearities are therefore present in the structure and the fluid problem have to be solved on a moving domain (2) the densities of the artery walls and the blood being close, the coupling is strong and has to be tackled very carefully to avoid numerical instabilities, (3) "naive" boundary conditions on the artificial boundaries induce spurious reflection phenomena.

Simulation of valves, either at the outflow of the cardiac chambers or in veins, is another example of difficult fluid-structure problems arising in blood flows. In addition, very large displacements and changes of topology (contact problems) have to be handled in those cases.

Because of the above mentioned difficulties, the interaction between the blood flow and the artery wall has often been neglected in most of the classical studies. The numerical properties of the fluid-structure coupling in blood flows are rather different from other classical fluid-structure problems. In particular, due to stability reasons it seems impossible to successfully apply the explicit coupling schemes used in aeroelasticity.

As a result, fluid-structure interaction in biological flows raise new challenging issues in scientific computing and numerical analysis : new schemes have to be developed and analyzed.

We have proposed over the last few years several efficient fluid-structure interaction algorithms. We are now using these algorithms to address inverse problems in blood flows (for example, estimation of artery wall stiffness from medical imaging).

3.1.2. Aerosol

Complex two-phase fluids can be modeled in many different ways. Eulerian models describe both phases by physical quantities such as the density, velocity or energy of each phase. In the mixed fluid-kinetic models, the biphasic fluid has one dispersed phase, which is constituted by a spray of droplets, with a possibly variable size, and a continuous classical fluid.

This type of model was first introduced by Williams [77] in the frame of combustion. It was later used to develop the Kiva code [67] at the Los Alamos National Laboratory, or the Hesione code [72], for example. It has a wide range of applications, besides the nuclear setting: diesel engines, rocket engines [70], therapeutic sprays, *etc.* One of the interests of such a model is that various phenomena on the droplets can be taken into account with an accurate precision: collision, breakups, coagulation, vaporization, chemical reactions, *etc.*, at the level of the droplets.

The model usually consists in coupling a kinetic equation, that describes the spray through a probability density function, and classical fluid equations (typically Navier-Stokes). The numerical solution of this system relies on the coupling of a method for the fluid equations (for instance, a finite volume method) with a method fitted to the spray (particle method, Monte Carlo).

We are mainly interested in modeling therapeutic sprays either for local or general treatments. The study of the underlying kinetic equations should lead us to a global model of the ambient fluid and the droplets, with some mathematical significance. Well-chosen numerical methods can give some tracks on the solutions behavior and help to fit the physical parameters which appear in the models.

3.2. Multiscale modeling

Multiscale modeling is a necessary step for blood and respiratory flows. In this section, we focus on blood flows. Nevertheless, similar investigations are currently carried out on respiratory flows.

3.2.1. Arterial tree modeling

Problems arising in the numerical modeling of the human cardiovascular system often require an accurate description of the flow in a specific sensible subregion (carotid bifurcation, stented artery, *etc.*). The description of such local phenomena is better addressed by means of three-dimensional (3D) simulations, based on the numerical approximation of the incompressible Navier-Stokes equations, possibly accounting for compliant (moving) boundaries. These simulations require the specification of boundary data on artificial boundaries that have to be introduced to delimit the vascular district under study. The definition of such boundary conditions is critical and, in fact, influenced by the global systemic dynamics. Whenever the boundary data is not available from accurate measurements, a proper boundary condition requires a mathematical description of the action of the reminder of the circulatory system on the local district. From the computational point of view, it is not affordable to describe the whole circulatory system keeping the same level of detail. Therefore, this mathematical description relies on simpler models, leading to the concept of *geometrical multiscale* modeling of the circulation [73]. The underlying idea consists in coupling different models (3D, 1D or 0D) with a decreasing level of accuracy, which is compensated by their decreasing level of computational complexity.

The research on this topic aims at providing a correct methodology and a mathematical and numerical framework for the simulation of blood flow in the whole cardiovascular system by means of a geometric multiscale approach. In particular, one of the main issues will be the definition of stable coupling strategies between 3D and reduced order models.

To model the arterial tree, a standard way consists of imposing a pressure or a flow rate at the inlet of the aorta, *i.e.* at the network entry. This strategy does not allow to describe important features as the overload in the heart caused by backward traveling waves. Indeed imposing a boundary condition at the beginning of the aorta artificially disturbs physiological pressure waves going from the arterial tree to the heart. The only way to catch this physiological behavior is to couple the arteries with a model of heart, or at least a model of left ventricle.

A constitutive law for the myocardium, controlled by an electrical command, has been developed in the CardioSense3D project [1]. One of our objectives is to couple artery models with this heart model.

A long term goal is to achieve 3D simulations of a system including heart and arteries. One of the difficulties of this very challenging task is to model the cardiac valves. To this purpose, we plan to mix arbitrary Lagrangian Eulerian and fictitious domain approaches, or simplified valve models based on an immersed surface strategy.

[1] http://www-sop.inria.fr/CardioSense3D/

3.2.2. Heart perfusion modeling

The heart is the organ that regulates, through its periodical contraction, the distribution of oxygenated blood in human vessels in order to nourish the different parts of the body. The heart needs its own supply of blood to work. The coronary arteries are the vessels that accomplish this task. The phenomenon by which blood reaches myocardial heart tissue starting from the blood vessels is called in medicine perfusion. The analysis of heart perfusion is an interesting and challenging problem. Our aim is to perform a three-dimensional dynamical numerical simulation of perfusion in the beating heart, in order to better understand the phenomena linked to perfusion. In particular the role of the ventricle contraction on the perfusion of the heart is investigated as well as the influence of blood on the solid mechanics of the ventricle. Heart perfusion in fact implies the interaction between heart muscle and blood vessels, in a sponge-like material that contracts at every heartbeat via the myocardium fibers.

Despite recent advances on the anatomical description and measurements of the coronary tree and on the corresponding physiological, physical and numerical modeling aspects, the complete modeling and simulation of blood flows inside the large and the many small vessels feeding the heart is still out of reach. Therefore, in order to model blood perfusion in the cardiac tissue, we must limit the description of the detailed flows at a given space scale, and simplify the modeling of the smaller scale flows by aggregating these phenomena into macroscopic quantities, by some kind of "homogenization" procedure. To that purpose, the modeling of the fluid-solid coupling within the framework of porous media appears appropriate.

Poromechanics is a simplified mixture theory where a complex fluid-structure interaction problem is replaced by a superposition of both components, each of them representing a fraction of the complete material at every point. It originally emerged in soils mechanics with the work of Terzaghi [76], and Biot [68] later gave a description of the mechanical behavior of a porous medium using an elastic formulation for the solid matrix, and Darcy's law for the fluid flow through the matrix. Finite strain poroelastic models have been proposed (see references in [69]), albeit with *ad hoc* formulations for which compatibility with thermodynamics laws and incompressibility conditions is not established.

3.2.3. Tumor and vascularization

The same way the myocardium needs to be perfused for the heart to beat, when it has reached a certain size, tumor tissue needs to be perfused by enough blood to grow. It thus triggers the creation of new blood vessels (angiogenesis) to continue to grow. The interaction of tumor and its micro-environment is an active field of research. One of the challenges is that phenomena (tumor cell proliferation and death, blood vessel adaptation, nutrient transport and diffusion, etc) occur at different scales. A multi-scale approach is thus being developed to tackle this issue. The long term objective is to predict the efficiency of drugs and optimize therapy of cancer.

3.2.4. Respiratory tract modeling

We aim to develop a multiscale modeling of the respiratory tract. Intraprenchymal airways distal from generation 7 of the tracheabronchial tree (TBT), which cannot be visualized by common medical imaging techniques, are modeled either by a single simple model or by a model set according to their order in TBT. The single model is based on straight pipe fully developed flow (Poiseuille flow in steady regimes) with given alveolar pressure at the end of each compartment. It will provide boundary conditions at the bronchial ends of 3D TBT reconstructed from imaging data. The model set includes three serial models. The generation down to the pulmonary lobule will be modeled by reduced basis elements. The lobular airways will be represented by a fractal homogenization approach. The alveoli, which are the gas exchange loci between blood and inhaled air, inflating during inspiration and deflating during expiration, will be described by multiphysics homogenization.

SISYPHE Project-Team

3. Scientific Foundations

3.1. System theory for systems modeled by ordinary differential equations

3.1.1. Identification, observation, control and diagnosis of linear and nonlinear systems

Characterizing and inferring properties and behaviors of objects or phenomena from observations using models is common to many research fields. For dynamical systems encountered in the domains of engineering and physiology, this is of practical importance for monitoring, prediction, and control. For such purposes, we consider most frequently, the following model of dynamical systems:

$$\begin{array}{rcl} \dfrac{dx(t)}{dt} & = & f(x(t),u(t),\theta,w(t)) \\ y(t) & = & g(x(t),u(t),\theta,v(t)) \end{array} \tag{1}$$

where $x(t)$, $u(t)$ and $y(t)$ represent respectively the state, input and output of the system, f and g characterize the state and output equations, parameterized by θ and subject to modeling and measurement uncertainties $w(t)$ and $v(t)$. Modeling is usually based on physical knowledge or on empirical experiences, strongly depending on the nature of the system. Typically only the input $u(t)$ and output $y(t)$ are directly observed by sensors. Inferring the parameters θ from available observations is known as system identification and may be useful for system monitoring [102], whereas algorithms for tracking the state trajectory $x(t)$ are called observers. The members of SISYPHE have gained important experiences in the modeling of some engineering systems and biomedical systems. The identification and observation of such systems often remain challenging because of strong nonlinearities [21]. Concerning control, robustness is an important issue, in particular to ensure various properties to all dynamical systems in some sets defined by uncertainties [83], [84]. The particularities of ensembles of connected dynamical systems raise new challenging problems.

Examples of reduced order models:
- Reduced order modeling of the cardiovascular system for signal & image processing or control applications. See section 3.3.1 .
- Excitable neuronal networks & control of the reproductive axis by the GnRH. See section 3.3.2 .
- Modeling, Control, Monitoring and Diagnosis of Depollution Systems. See section 6.1 .

3.2. System theory for quantum and quantum-like systems

3.2.1. Quantization of waves propagation in transmission-line networks & Inverse scattering

Linear stationary waves. Our main example of classical system that is interesting to see as a quantum-like system is the Telegrapher Equation, a model of transmission lines, possibly connected into a network. This is the standard model for electrical networks, where V and I are the voltage and intensity functions of z and k, the position and frequency and $R(z)$, $L(z)$, $C(z)$, $G(z)$ are the characteristics of the line:

$$\frac{\partial V(z,k)}{\partial z} = -\left(R(z)+jkL(z)\right)I(z,k), \quad \frac{\partial I(z,k)}{\partial z} = -(G(z)+jkC(z))V(z,k) \tag{2}$$

Since the work of Noordergraaf [101], this model is also used for hemodynamic networks with V and I respectively the blood pressure and flow in vessels considered as 1D media, and with $R=\frac{8\pi\eta}{S^2}$, $L=\frac{\rho}{S}$, $C=\frac{3S(r+h)}{E(2r+h)}$ where ρ and η are the density and viscosity of the blood ; r, h and E are the inner radius, thickness and Young modulus of the vessel. $S=\pi r^2$. The conductivity G is a small constant for blood flow.

Monitoring such networks is leading us to consider the following inverse problem: *get information on the functions R, L, C, G from the reflection coefficient* $\mathcal{R}(k)$ (ratio of reflected over direct waves) measured in some location by Time or Frequency Domain Reflectometry.

To study this problem it is convenient to use a Liouville transform, setting $x(z) = \int_0^z \sqrt{L(z')C(z')}dz'$, to introduce auxiliary functions $q^{\pm}(x) = \frac{1}{4}\frac{d}{x}\left(\ln\frac{L(x)}{C(x)}\right) \pm \frac{1}{2}\left(\frac{R(x)}{L(x)} - \frac{G(x)}{C(x)}\right)$ and $q_p(x) = \frac{1}{2}\left(\frac{R(x)}{L(x)} + \frac{G(x)}{C(x)}\right)$, so that (2) becomes a Zakharov-Shabat system [89] that reduces to a Schrödinger equation in the lossless case ($R = G = 0$):

$$\frac{\partial v_1}{\partial x} = (q_p - jk)\, v_1 + q^+ v_2, \quad \frac{\partial v_2}{\partial x} = -(q_p - jk)\, v_2 + q^- v_1 \tag{3}$$

and $I(x,k) = \frac{1}{\sqrt{2}}\left[\frac{C(x)}{L(x)}\right]^{\frac{1}{4}}(v_1(x,k) + v_2(x,k))$, $V(x,k) = -\frac{1}{\sqrt{2}}\left[\frac{L(x)}{C(x)}\right]^{\frac{1}{4}}(v_1(x,k) - v_2(x,k))$.
Our inverse problem becomes now an inverse scattering problem for a Zakharov-Shabat (or Schrödinger) equation: *find the potentials $q^{\pm}$ and q_p corresponding to* $\mathcal{R}$. This classical problem of mathematical physics can be solved using e.g. the Gelfand-Levitan-Marchenko method.

Nonlinear traveling waves. In some recent publications [92], [91], we use scattering theory to analyze a measured Arterial Blood Pressure (ABP) signal. Following a suggestion made in [103], a Korteweg-de Vries equation (KdV) is used as a physical model of the arterial flow during the pulse transit time. The signal analysis is based on the use of the Lax formalism: the iso-spectral property of the KdV flow allows to associate a constant spectrum to the non stationary signal. Let the non-dimensionalized KdV equation be

$$\frac{\partial y}{\partial t} - 6y\frac{\partial y}{\partial x} + \frac{\partial^3 y}{\partial x^3} = 0 \tag{4}$$

In the Lax formalism, y is associated to a Lax pair: a Schrödinger operator, $L(y) = -\frac{\partial^2}{\partial x^2} + y$ and an anti-Hermitian operator $M(y) = -4\frac{\partial^3}{\partial x^3} + 3y\frac{\partial}{\partial x} + 3\frac{\partial}{\partial x}y$. The signal y is playing here the role of the potential of $L(y)$ and is given by an operator equation equivalent to (4):

$$\frac{\partial L(y)}{\partial t} = [M(y), L(y)] \tag{5}$$

Scattering and inverse scattering transforms can be used to analyze y in term of the spectrum of $L(y)$ and conversely. The "bound states" of $L(y)$ are of particular interest: if $L(y)$ is solution of (5) and $L(y(t))$ has only bound states (no continuous spectrum), then this property is true at each time and y is a soliton of KdV. For example the arterial pulse pressure is close to a soliton [86], [14].

Inverse scattering as a generalized Fourier transform. For "pulse-shaped" signals y, meaning that $y \in L^1(\mathbb{R}; (1+|x|^2)dx)$, the squared eigenfunctions of $L(y)$ and their space derivatives are a basis in $L^1(\mathbb{R}; dx)$ (see e.g. [99]) and we use this property to analyze signals. Remark that the Fourier transform corresponds to using the basis associated with $L(0)$. The expression of a signal y in its associated basis is of particular interest. For a positive signal (as e.g. the arterial pressure), it is convenient to use $L(-y)$ as $-y$ is like a multi-well potential, and the Inverse scattering transform formula becomes:

$$y(x) = 4\sum_{n=1}^{n=N}\kappa_n\psi_n^2(x) - \frac{2i}{\pi}\int_{-\infty}^{-\infty} k\mathcal{R}(k)f^2(k,x)dk \tag{6}$$

where ψ_n and $f(k,.)$ are solutions of $L(-y)f = k^2 f$ with $k = i\kappa_n$, $\kappa_n > 0$, for ψ_n (bound states) and $k > 0$ for $f(k,.)$ (Jost solutions). The discrete part of this expression is easy to compute and provides useful informations on y in applications. The case $\mathcal{R} = 0$ ($-y$ is a reflectionless potential) is then of particular interest as $2N$ parameters are sufficient to represent the signal. We investigate in particular approximation of pulse-shaped signals by such potentials corresponding to N-solitons.

3.2.2. Identification & control of quantum systems

Interesting applications for quantum control have motivated seminal studies in such wide-ranging fields as chemistry, metrology, optical networking and computer science. In chemistry, the ability of coherent light to manipulate molecular systems at the quantum scale has been demonstrated both theoretically and experimentally [98]. In computer science, first generations of quantum logical gates (restrictive in fidelity) has been constructed using trapped ions controlled by laser fields (see e.g. the "Quantum Optics and Spectroscopy Group, Univ. Innsbruck"). All these advances and demands for more faithful algorithms for manipulating the quantum particles are driving the theoretical and experimental research towards the development of new control techniques adapted to these particular systems. A very restrictive property, particular to the quantum systems, is due to the destructive behavior of the measurement concept. One can not measure a quantum system without interfering and perturbing the system in a non-negligible manner.

Quantum decoherence (environmentally induced dissipations) is the main obstacle for improving the existing algorithms [88]. Two approaches can be considered for this aim: first, to consider more resistant systems with respect to this quantum decoherence and developing faithful methods to manipulate the system in the time constants where the decoherence can not show up (in particular one can not consider the back-action of the measurement tool on the system); second, to consider dissipative models where the decoherence is also included and to develop control designs that best confronts the dissipative effects.

In the first direction, we consider the Schrödinger equation where $\Psi(t,x)$, $-\frac{1}{2}\Delta$, V, μ and $u(t)$ respectively represent the wavefunction, the kinetic energy operator, the internal potential, the dipole moment and the laser amplitude (control field):

$$i\frac{d}{dt}\Psi(t,x) = (H_0 + u(t)H_1)\Psi(t,x) = (-\frac{1}{2}\Delta + V(x) + u(t)\mu(x))\Psi(t,x), \quad \Psi|_{t=0} = \Psi_0, \tag{7}$$

While the finite dimensional approximations ($\Psi(t) \in \mathbb{C}^N$) have been very well studied (see e.g. the works by H. Rabitz, G. Turinici, ...), the infinite dimensional case ($\Psi(t,.) \in L^2(\mathbb{R}^N;\mathbb{C})$) remains fairly open. Some partial results on the controllability and the control strategies for such kind of systems in particular test cases have already been provided [79], [80], [94]. As a first direction, in collaboration with K. Beauchard (CNRS, ENS Cachan) et J-M Coron (Paris-sud), we aim to extend the existing ideas to more general and interesting cases. We will consider in particular, the extension of the Lyapunov-based techniques developed in [95], [81], [94]. Some technical problems, like the pre-compactness of the trajectories in relevant functional spaces, seem to be the main obstacles in this direction.

In the second direction, one needs to consider dissipative models taking the decoherence phenomena into account. Such models can be presented in the density operator language. In fact, to the Schrödinger equation (7), one can associate an equation in the density operator language where $\rho = \Psi\Psi^*$ represents the projection operator on the wavefunction Ψ ($[A,B] = AB - BA$ is the commutator of the operators A and B):

$$\frac{d}{dt}\rho = -i[H_0 + u(t)H_1, \rho], \tag{8}$$

Whenever, we consider a quantum system in its environment with the quantum jumps induced by the vacuum fluctuations, we need to add the dissipative effect due to these stochastic jumps. Note that at this level, one also can consider a measurement tool as a part of the environment. The outputs being partial and not giving complete information about the state of the system (Heisenberg uncertainty principle), we consider a so-called quantum filtering equation in order to model the conditional evolution of the system. Whenever the measurement tool composes the only (or the only non-negligible) source of decoherence, this filter equation admits the following form:

$$\begin{aligned} d\rho_t = -i[H_0 + u(t)H_1, \rho_t]dt + (L\rho_t L^* - \frac{1}{2}L^*L\rho_t - \frac{1}{2}\rho_t L^* L)dt \\ +\sqrt{\eta}(L\rho_t + \rho_t L^* - \mathrm{Tr}\left[(L+L^*)\rho_t\right]\rho_t)dW_t, \end{aligned} \tag{9}$$

where L is the so-called Lindblad operator associated to the measurement, $0 < \eta \leq 1$ is the detector's efficiency and where the Wiener process W_t corresponds to the system output Y_t via the relation $dW_t = dY_t - \mathrm{Tr}\left[(L+L^*)\rho_t\right]dt$. This filter equation, initially introduced by Belavkin [82], is the quantum analogous of a Kushner-Stratonovic equation. In collaboration with H. Mabuchi and his co-workers (Physics department, Caltech), we would like to investigate the derivation and the stochastic control of such filtering equations for different settings coming from different experiments [96].

Finally, as a dual to the control problem, physicists and chemists are also interested in the parameter identification for these quantum systems. Observing different physical observables for different choices of the input $u(t)$, they hope to derive more precise information about the unknown parameters of the system being parts of the internal Hamiltonian or the dipole moment. In collaboration with C. Le Bris (Ecole des ponts and Inria), G. Turinici (Paris Dauphine and Inria), P. Rouchon (Ecole des Mines) and H. Rabitz (Chemistry department, Princeton), we would like to propose new methods coming from the systems theory and well-adapted to this particular context. A first theoretical identifiability result has been proposed [93]. Moreover, a first observer-based identification algorithm is under study.

3.3. Physiological & Clinical research topics

3.3.1. The cardiovascular system: a multiscale controlled system

Understanding the complex mechanisms involved in the cardiac pathological processes requires fundamental researches in molecular and cell biology, together with rigorous clinical evaluation protocols on the whole organ or system scales. Our objective is to contribute to to these researches by developing low-order models of the cardiac mechano-energetics and control mechanisms, for applications in model-based cardiovascular signal or image processing.

We consider intrinsic heart control mechanisms, ranging from the Starling and Treppe effects on the cell scale to the excitability of the cardiac tissue and to the control by the autonomous nervous system. They all contribute to the function of the heart in a coordinated manner that we want to analyze and assess. For this purpose, we study reduced-order models of the electro-mechanical activity of cardiac cells designed to be coupled with measures available on the organ scale (e.g. ECG and pressure signals). We study also the possibility to gain insight on the cell scale by using model-based multiscale signal processing techniques of long records of cardiovascular signals.

Here are some questions of this kind, we are considering:
- Modeling the controlled contraction/relaxation from molecular to tissue and organ scales.
- Direct and inverse modeling the electro-mechanical activity of the heart on the cell scale.
- Nonlinear spectral analysis of arterial blood pressure waveforms and application to clinical indexes.
- Modeling short-term and long-term control dynamics on the cardiovascular-system scale. Application to a Total Artificial Heart.

3.3.2. Reproductive system: follicular development & ovulation control

The ovulatory success is the main limiting factor of the whole reproductive process, so that a better understanding of ovulation control is needed both for clinical and zootechnical applications. It is necessary to improve the treatment of anovulatory infertility in women, as it can be by instance encountered in the PolyCystic Ovarian Syndrome (PCOS), whose prevalence among reproductive-age women has been estimated at up to 10%. In farm domestic species, embryo production following FSH stimulation (and subsequent insemination) enables to amplify the lineage of chosen females (via embryo transfer) and to preserve the genetic diversity (via embryo storage in cryobanks). The large variability in the individual responses to ovarian stimulation treatment hampers both their therapeutic and farming applications. Improving the knowledge upon the mechanisms underlying FSH control will help to improve the success of assisted reproductive technologies, hence to prevent ovarian failure or hyperstimulation syndrome in women and to manage ovulation rate and ovarian cycle chronology in farm species.

To control ovarian cycle and ovulation, we have to deeply understand the selection process of ovulatory follicles, the determinism of the species-specific ovulation rate and of its intra- and between-species variability, as well as the triggering of the ovulatory GnRH surge from hypothalamic neurons.
Beyond the strict scope of Reproductive Physiology, this understanding raises biological questions of general interest, especially in the fields of

Molecular and Cellular Biology. The granulosa cell, which is the primary target of FSH in ovarian follicles, is a remarkable cellular model to study the dynamical control of the transitions between the cellular states of quiescence, proliferation, differentiation, and apoptosis, as well as the adaptability of the response to the same extra-cellular signal according to the maturity level of the target cell. Moreover, the FSH receptor belongs to the seven transmembrane spanning receptor family, which represent the most frequent target (over 50%) amongst the therapeutic agents currently available. The study of FSH receptor-mediated signaling is thus not only susceptible to allow the identification of relaying controls to the control exerted by FSH, but it is also interesting from a more generic pharmacological viewpoint.

Neuroendocrinology and Chronobiology. The mechanisms underlying the GnRH ovulatory surge involve plasticity phenomena of both neuronal cell bodies and synaptic endings comparable to those occurring in cognitive processes. Many time scales are interlinked in ovulation control from the fastest time constants of neuronal activation (millisecond) to the circannual variations in ovarian cyclicity. The influence of daylength on ovarian activity is an interesting instance of a circannual rhythm driven by a circadian rhythm (melatonin secretion from the pineal gland).

Simulation and control of a multiscale conservation law for follicular cells

In the past years, we have designed a multiscale model of the selection process of ovulatory follicles, including the cellular, follicular and ovarian levels [11], [10]. The model results from the double structuration of the granulosa cell population according to the cell age (position within the cell cycle) and to the cell maturity (level of sensitivity towards hormonal control). In each ovarian follicle, the granulosa cell population is described by a density function whose changes are ruled by conservation laws. The multiscale structure arises from the formulation of a hierarchical control operating on the aging and maturation velocities as well on the source terms of the conservation law. The control is expressed from different momentums of the density leading to integro-differential expressions.
Future work will take place in the REGATE project and will consist in:
- predicting the selection outcome (mono-, poly-ovulation or anovulation / ovulation chronology) resulting from given combinations of parameters and corresponding to the subtle interplay between the different organs of the gonadotropic axis (hypothalamus, pituitary gland and ovaries). The systematic exploration of the situations engendered by the model calls for the improvement of the current implementation performances. The work will consist in improving the precision of the numerical scheme, in the framework of the finite volume method and to implement the improved scheme,

- solving the control problems associated with the model. Indeed, the physiological conditions for the triggering of ovulation, as well as the counting of ovulatory follicles amongst all follicles, define two nested and coupled reachability control problems. Such particularly awkward problems will first be tackled from a particle approximation of the density, in order to design appropriate control laws operating on the particles and allowing them to reach the target state sets.

Connectivity and dynamics of the FSH signaling network in granulosa cells

The project consists in analyzing the connectivity and dynamics of the FSH signaling network in the granulosa cells of ovarian follicles and embedding the network within the multiscale representation described above, from the molecular up to the organic level. We will examine the relative contributions of the Gα_s and βarrestin-dependent pathways in response to FSH signal, determine how each pathway controls downstream cascades and which mechanisms are involved in the transition between different cellular states (quiescence, proliferation, differentiation and apoptosis). On the experimental ground, we propose to develop an antibody microarray approach in order to simultaneously measure the phosphorylation levels of a large number of signaling intermediates in a single experiment. On the modeling ground, we will use the BIOCHAM (biochemical abstract machine) environment first at the boolean level, to formalize the network of interactions corresponding to the FSH-induced signaling events on the cellular scale. This network will then be enriched with kinetic information coming from experimental data, which will allow the use of the ordinary differential equation level of BIOCHAM. In order to find and fine-tune the structure of the network and the values of the kinetic parameters, model-checking techniques will permit a systematic comparison between the model behavior and the results of experiments. In the end, the cell-level model should be abstracted to a much simpler model that can be embedded into a multiscale one without losing its main characteristics.

Bifurcations in coupled neuronal oscillators.

We have proposed a mathematical model allowing for the alternating pulse and surge pattern of GnRH (Gonadotropin Releasing Hormone) secretion [5]. The model is based on the coupling between two systems running on different time scales. The faster system corresponds to the average activity of GnRH neurons, while the slower one corresponds to the average activity of regulatory neurons. The analysis of the slow/fast dynamics exhibited within and between both systems allows to explain the different patterns (slow oscillations, fast oscillations and periodical surge) of GnRH secretion.

This model will be used as a basis to understand the control exerted by ovarian steroids on GnRH secretion, in terms of amplitude, frequency and plateau length of oscillations and to discriminate a direct action (on the GnrH network) from an indirect action (on the regulatory network) of steroids. From a mathematical viewpoint, we have to fully understand the sequences of bifurcations corresponding to the different phases of GnRH secretion. This study will be derived from a 3D reduction of the original model.

ARLES Project-Team

3. Scientific Foundations

3.1. Introduction

Research undertaken within the ARLES project-team aims to offer comprehensive solutions to support the development of pervasive computing systems that are dynamically composed according to networked resources in the environment. This leads us to investigate methods and tools supporting the engineering of pervasive software systems, with a special emphasis on associated middleware solutions.

3.2. Engineering Pervasive Software Systems

Since its emergence, middleware has proved successful in assisting distributed software development, making development faster and easier, and significantly promoting software reuse while overcoming the heterogeneity of the distributed infrastructure. As a result, middleware-based software engineering is central to the principled development of pervasive computing systems. In this section, we (i) discuss challenges that middleware brings to software engineering, and (ii) outline a revolutionary approach to middleware-based software engineering aiming at the dynamic runtime synthesis of emergent middleware.

3.2.1. Middleware-based Software Engineering

Middleware establishes a new software layer that homogenizes the infrastructure's diversities by means of a well-defined and structured distributed programming model, relieving software developers from low-level implementation details, by: (i) at least abstracting transport layer network programming via high-level network abstractions matching the application computational model, and (ii) possibly managing networked resources to offer quality of service guarantees and/or domain specific functionalities, through reusable middleware-level services. More specifically, middleware defines:

- A resource definition language that is used for specifying data types and interfaces of networked software resources;
- A high-level addressing scheme based on the underlying network addressing scheme for locating resources;
- Interaction paradigms and semantics for achieving coordination;
- A transport/session protocol for achieving communication; and
- A naming/discovery protocol with related registry structure and matching relation for publishing and discovering the resources available in the given network.

Attractive features of middleware have made it a powerful tool in the software system development practice. Hence, middleware is a key factor that has been and needs to be further taken into account in the Software Engineering (SE) discipline [5]. The advent of middleware standards have further contributed to the systematic adoption of this paradigm for distributed software development.

[5] W. Emmerich. Software Engineering and Middleware: a roadmap. In Proceedings of the Conference on the Future of Software Engineering, Limerick, Ireland, Jun. 2000.

In spite of the above, mature engineering methodologies to comprehensively assist the development of middleware-based software systems, from requirements analysis to deployment and maintenance, are lagging behind. Indeed, systematic software development accounting for middleware support is rather the exception than the norm, and methods and related tools are dearly required for middleware-based software engineering. This need becomes even more demanding if we consider the diversity and scale of today's networking environments and application domains, which makes middleware and its association with applications highly complex [5], raising new, challenging requirements for middleware. Among those, access to computational resources should be open across network boundaries and dynamic due to the potential mobility of host- and user-nodes. This urges middleware to support methods and mechanisms for description, dynamic discovery and association, late binding, and loose coordination of resources. In such variable and unpredictable environments, operating not only according to explicit system inputs but also according to the context of system operation becomes of major importance, which should be enabled by the middleware. Additionally, the networking infrastructure is continuing to evolve at a fast pace, and suggesting new development paradigms for distributed systems, calling for next-generation middleware platforms and novel software engineering processes integrating middleware features in all phases of the software development.

3.2.2. Beyond Middleware-based Architectures for Interoperability

As discussed above, middleware stands as the conceptual paradigm to effectively network together heterogeneous systems, specifically providing upper layer interoperability. That said, middleware is yet another technological block, which creates islands of networked systems.

Interoperable middleware has been introduced to overcome middleware heterogeneity. However, solutions remain rather static, requiring either use of a proprietary interface or a priori implementation of protocol translators. In general, interoperability solutions solve protocol mismatch among middleware at syntactic level, which is too restrictive. This is even truer when one considers the many dimensions of heterogeneity, including software, hardware and networks, which are currently present in ubiquitous networking environments, and that require fine tuning of the middleware according to the specific capacities embedded within the interacting parties. Thus, interoperable middleware can at best solve protocol mismatches arising among middleware aimed at a specific domain. Indeed, it is not possible to a priori design a universal middleware solution that will enable effective networking of digital systems, while spanning the many dimensions of heterogeneity currently present in networked environments and further expected to increase dramatically in the future.

A revolutionary approach to the seamless networking of digital systems is to synthesize connectors on the fly, via which networked systems communicate. The resulting emergent connectors then compose and further adapt the interaction protocols run by the connected systems, from the application layer down to the middleware layer. Hence, thanks to results in this new area, networked digital systems will survive the obsolescence of interaction protocols and further emergence of new ones.

We have specifically undertaken cooperative research on the dynamic synthesis of emergent connectors which shall rely on a formal foundation for connectors that allows learning, reasoning about, and adapting the interaction behavior of networked systems [6]. Further, compared to the state of the art foundations for connectors, it should operate a drastic shift by learning, reasoning about, and synthesizing connector behavior at run-time. Indeed, the use of connector specifications pioneered by the software architecture research field has mainly been considered as a design-time concern, for which automated reasoning is now getting practical even if limitations remain. On the other hand, recent effort in the semantic Web domain brings ontology-based semantic knowledge and reasoning at run-time; however, networked system solutions based thereupon are currently mainly focused on the functional behavior of networked systems, with few attempts to capture their interaction behavior as well as non-functional properties. In this new approach, the interaction protocols (both application- and middleware-layer) behavior will be learnt by observing the interactions of the networked

[6] Valérie Issarny, Bernhard Steffen, Bengt Jonsson, Gordon S. Blair, Paul Grace, Marta Z. Kwiatkowska, Radu Calinescu, Paola Inverardi, Massimo Tivoli, Antonia Bertolino, Antonino Sabetta: CONNECT Challenges: Towards Emergent Connectors for Eternal Networked Systems. In Proceedings of ICECCS 2009.

systems, where ontology-based specification and other semantic knowledge will be exploited for generating connectors on the fly.

3.3. Middleware Architectures for Pervasive Computing

Today's wireless networks enable dynamically setting up temporary networks among mobile nodes for the realization of some distributed function. However, this requires adequate development support and, in particular, supporting middleware platforms for alleviating the complexity associated with the management of dynamic networks composed of highly heterogeneous nodes. In this section, we present an overview of: (i) service oriented middleware, a prominent paradigm in large distributed systems today, and (ii) middleware for wireless sensor networks, which have recently emerged as a promising platform.

3.3.1. Service Oriented Middleware

The *Service Oriented Computing*(SOC) paradigm advocates that networked resources should be abstracted as services, thus allowing their open and dynamic discovery, access and composition, and hence reuse. Due to this flexibility, SOC has proven to be a key enabler for pervasive computing. Moreover, SOC enables integrating pervasive environments into broader service oriented settings: the current and especially the *Future Internet* is the ultimate case of such integration. We, more particularly, envision the Future Internet as a ubiquitous setting where services representing resources, people and things can be freely and dynamically composed in a decentralized fashion, which is designated by the notion of service choreography in the SOC idiom. In the following, we discuss the role that *service oriented middleware* is aimed to have within our above sketched vision of the Future Internet, of which pervasive computing forms an integral part.

From service oriented computing to service oriented middleware: In the last few years, there is a growing interest in choreography as a key concept in forming complex service-oriented systems. Choreography is put forward as a generic abstraction of any possible collaboration among multiple services, and integrates previously established views on service composition, among which service orchestration. Several different approaches to choreography modeling can be found in the literature: *Interaction-oriented* models describe choreography as a set of interactions between participants; while *process-oriented* models describe choreography as a parallel composition of the participants' business processes. *Activity-based* models focus on the interactions between the parties and their ordering, whereas the state of the interaction is not explicitly modeled or only partly modeled using variables; while *state-based* models model the states of the choreography as first-class entities, and the interactions as transitions between states.

The above modeling categorizations are applied in the ways in which: service choreographies are specified (e.g., by employing languages such as BPMN, WS-CDL, BPEL); services are discovered, selected and composed into choreographies (e.g., based on their features concerning interfaces, behavior, and non-functional properties such as QoS and context); heterogeneity between choreographed services is resolved via adaptation (e.g., in terms of service features and also underlying communication protocols); choreographies are deployed and enacted (e.g., in terms of deployment styles and execution engines); and choreographies are maintained/adapted given the independent evolution of choreographed services (e.g., in terms of availability and QoS). These are demanding functionalities that service oriented middleware should provide for supporting service choreographies. In providing these functionalities in the context of the Future Internet, service oriented middleware is further challenged by two key Future Internet properties: its *ultra large scale* as in number of users and services, and the *high degree of heterogeneity* of services, whose hosting platforms may range from that of resource-rich, fixed hosts to wireless, resource-constrained devices. These two properties call for considerable advances to the state of the art of the SOC paradigm.

Our work in the last years has focused on providing solutions to the above identified challenges, more particularly in the domain of pervasive computing. Given the prevalence of mobile networking environments and powerful hand-held consumer devices, we consider resource constrained devices (and things, although we focus on smart, i.e., computation-enabled, things) as first-class entities of the Future Internet. Concerning middleware that enables networking mobile and/or resource constrained devices in pervasive computing environments, several promising solutions have been proposed, such as mobile Gaia, TOTA, AlfredO, or work

at UCL, Carnegie Mellon University, and the University of Texas at Arlington. They address issues such as resource discovery, resource access, adaptation, context awareness as in location sensitivity, and pro-activeness in a seamless manner. Other solutions specialize in sensor networks; we, more specifically, discuss middleware for wireless sensor networks in the next section. In this very active domain of service-oriented middleware for pervasive computing environments, we have extensive expertise that ranges from lower-level cross-layer networking to higher-level semantics of services, as well as transversal concerns such as context and privacy. We have in particular worked on aspects including semantic discovery and composition of services based on their functional properties, heterogeneity of service discovery protocols, and heterogeneity of network interfaces. Based on our accumulated experience, we are currently focusing on some of the still unsolved challenges identified above.

QoS-aware service composition: With regard to service composition in pervasive environments, taking into account QoS besides functional properties ensures a satisfactory experience to the end user. We focus here on the orchestration-driven case, where service composition is performed to fulfill a task requested by the user along with certain QoS constraints. Assuming the availability of multiple resources in service environments, a large number of services can be found for realizing every sub-task part of a complex task. A specific issue emerges in this regard, which is about selecting the best set of services (i.e., in terms of QoS) to participate in the composition, meeting user's global QoS requirements. QoS-aware composition becomes even more challenging when it is considered in the context of dynamic service environments characterized by changing conditions. As dynamic environments call for fulfilling user requests on the fly (i.e., at run-time) and as services' availability cannot be known a priori, service selection and composition must be performed at runtime. Hence, the execution time of service selection algorithms is heavily constrained, whereas the computational complexity of this problem is NP-hard.

Coordination of heterogeneous distributed systems: Another aspect that we consider important in service composition is enabling integration of services that employ different interaction paradigms. Diversity and ultra large scale of the Future Internet have a direct impact on coordination among interacting entities. Our choice of choreography as global coordination style among services should further be underpinned by support for and interoperability between heterogeneous interaction paradigms, such as message-driven, event-driven and data-driven ones. Different interaction paradigms apply to different needs: for instance, asynchronous, event-based publish/subscribe is more appropriate for highly dynamic environments with frequent disconnections of involved entities. Enabling interoperability between such paradigms is imperative in the extremely heterogeneous Future Internet integrating services, people and things. Interoperability efforts are traditionally based on, e.g., bridging communication protocols, where the dominant position is held by ESBs, wrapping systems behind standard technology interfaces, and/or providing common API abstractions. However, such efforts mostly concern a single interaction paradigm and thus do not or only poorly address cross-paradigm interoperability. Efforts combining diverse interaction paradigms include: implementing the LIME tuple space middleware on top of a publish/subscribe substrate; enabling Web services/SOAP-based interactions over a tuple space binding; and providing ESB implementations based on the tuple space paradigm.

Evolution of service oriented applications: A third issue we are interested in concerns the maintenance of service-oriented applications despite the evolution of employed services. Services are autonomous systems that have been developed independently from each other. Moreover, dynamics of pervasive environments and the Future Internet result in services evolving independently; a service may be deployed, or un-deployed at anytime; its implementation, along with its interface may change without prior notification. In addition, there are many evolving services that offer the same functionality via different interfaces and with varying quality characteristics (e.g., performance, availability, reliability). The overall maintenance process amounts to replacing a service that no longer satisfies the requirements of the employing application with a substitute service that offers the same or a similar functionality. The goal of seamless service substitution is to relate the substitute service with the original service via concrete mappings between their operations, their inputs and outputs. Based on such mappings, it is possible to develop/generate an adapter that allows the employing application to access the substitute service without any modification in its implementation. The service substitution should be dynamic and efficient, supported by a high level of automation. The state

of the art in service substitution comprises various approaches. There exist efforts, which assume that the mappings between the original and the substitute service are given, specified by the application or the service providers. The human effort required makes these approaches impractical, especially in the case of pervasive environments. On the other hand, there exist automated solutions, proposing mechanisms for the derivation of mappings. The complexity of these approaches scales up with the cardinality of available services and therefore efficiency is compromised. Again, this is an important disadvantage, especially considering the case of pervasive environments.

3.3.2. Middleware for Wireless Sensor Networks

Wireless sensor networks (WSNs) enable low cost, dense monitoring of the physical environment through collaborative computation and communication in a network of autonomous sensor nodes, and are an area of active research. Owing to the work done on system-level functionalities such as energy-efficient medium access and data-propagation techniques, sensor networks are being deployed in the real world, with an accompanied increase in network sizes, amount of data handled, and the variety of applications. The early networked sensor systems were programmed by the scientists who designed their hardware, much like the early computers. However, the intended developer of sensor network applications is not the computer scientist, but the designer of the system *using* the sensor networks, which might be deployed in a building or a highway. We use the term *domain expert* to mean the class of individuals most likely to use WSNs – people who may have basic programming skills but lack the training required to program distributed systems. Examples of domain experts include architects, civil and environmental engineers, traffic system engineers, medical system designers etc. We believe that the wide acceptance of networked sensing is dependent on the ease-of-use experienced by the domain expert in developing applications on such systems.

The obvious solution to enable this ease-of-use in application development is sensor network middleware, along with related programming abstractions [7]. Recent efforts in standardizing network-layer protocols for embedded devices provide a sound foundation for research and development of middleware that assist the sensor network developers in various aspects that are of interest to us, including the following.

Data-oriented operations: A large number of WSN applications are concerned with sampling and collection of data, and this has led to a large body of work to provide middleware support to the programmer of WSNs for easy access to the data generated and needed by the constituent nodes. Initial work included Hood, and TeenyLIME, which allowed data-sharing over a limited spatial range. Further work proposed the use of the DART runtime environment, which exposes the sensor network as a distributed data-store, addressable by using logical addresses such as “all nodes with temperature sensors in Room 503”, or “all fire sprinklers in the fifth and sixth floors”, which are more intuitive than, say, IP addresses. Taking a different approach toward handling the data in the sensor network, some middleware solutions propose to manipulate them using semantic techniques, such as in the Triple Space Computing approach, which models the data shared by the nodes in the system as RDF triples (subject-predicate-object groups), a standard method for semantic data representation. They propose to make these triples available to the participating nodes using a tuple space, thus giving it the “triple space” moniker. S-APL or Semantic-Agent Programming Language uses semantic technologies to integrate the semantic descriptions of the domain resources with the semantic prescription of agent behavior.

Integration with non-WSN nodes: Most of the work above focuses on designing applications that exhibit only intra-network interactions, where the interaction with the outside world is only in the form of sensing it, or controlling it by actuation. The act of connecting this data to other systems outside the sensor network is mostly done using an external gateway. This is then supported by middlewares that expose the sensor network as a database (e.g., TinyDB and Cougar), allowing the operator to access the data using a SQL-like syntax, augmented with keywords that can be used to specify the rate of sampling, for example. Another direction of integrating WSNs in general with larger systems such as Web servers has been toward using REST (REpresentational State Transfer) technologies, which are already used for accessing services on the Web as a

[7] L. Mottola and G. P. Picco. Programming Wireless Sensor Networks: Fundamental Concepts and State of the Art. In ACM Computing Surveys. Volume 43, Issue 3. April 2011.

lightweight alternative to SOAP. There has also been work proposing a system that will enable heterogeneous sensor and actuators to expose their sensing and actuation capabilities in a plug and play fashion. It proposes a middleware that defines a set of constraints, support services and interaction patterns that follow the REST architectural style principles, using the ATOM Web publishing protocol for service description, and a two-step discovery process. Additionally, there has been work in implementation of a REST-oriented middleware that runs on embedded devices such as Sun SPOT nodes, and the Plogg wireless energy monitors. This involves a two-fold approach — embedding tiny Web servers in devices that can host them, and employing a proxy server in situations where that is not the case. However, it has been noticed that the abstractions provided by REST might be too simplistic to compose complex applications over the services provided by WSN nodes. Some of the most recent work in this area also proposes to convert existing (network-layer) gateways into smart gateways, by running application code on them.

In addition to supporting the above interactions, sensor network middleware has also been proposed to address the challenges arising from the fact that a particular sensor or actuator may not be always available. This leads to the need for transparent reconfiguration, where the application developer should not have to care about reliability issues. The PIRATES event-based middleware for resource-rich nodes (hosting sensors/actuators, or just processing data) includes a third-party-remapping facility that can be used to remap a component's endpoints without affecting the business logic. In that sense, it is similar to the RUNES middleware targeted at embedded systems.

Finally, we also note the recent initial WSN middleware research focused on the new nascent classes of systems. Most recently, the field of *participatory sensing*[8] has emerged, where the role of sensing is increasingly being performed by the mobile phones carried by the users of the system, providing data captured using the sound, GPS, accelerometer and other sensors attached to them. This has led to the emergence of middleware such as JigSaw. The core additional challenges in this domain come from the inherent mobility of the nodes, as well as their extremely large scale.

[8]Lane, N.D.; Miluzzo, E.; Hong Lu; Peebles, D.; Choudhury, T.; Campbell, A.T.; , "A survey of mobile phone sensing," Communications Magazine, IEEE , vol.48, no.9, pp.140-150, Sept. 2010

GANG Project-Team (section vide)

HIPERCOM Project-Team

3. Scientific Foundations

3.1. Analytical information theory

Participants: Cédric Adjih, Pascale Minet, Paul Mühlethaler.

channel capacity, compression, predictors

Information theory Branch of mathematics dedicated to the quantification of the performance of a medium to carry information. Initiated by Shannon in 1948.

Abstract. Information theory and analytical methods play a central role in the networking technology. It identifies the key parameter that must be quantified in order to characterize the performance of a network.

The analytical information theory is part of the foundations of the Hipercom project. This is a tool box that has been collected and adapted from the areas of the analysis of algorithms and the information theory. It provides powerful tool for the analysis of telecommunication algorithms. The analysis of the behavior of such algorithms in their asymptotic range are fundamental in order to identify their critical parts. It helps to design and properly scale the protocols. Application of analytical information theory ranges from channel capacity computations, compression algorithm performance evaluation, predictor designs.

3.2. Methodology of telecommunication algorithm evaluation

Participants: Cédric Adjih, Ichrak Amdouni, Emmanuel Baccelli, Salman Malik, Yacine Mezali, Pascale Minet, Paul Mühlethaler, Saoucene Mahfoudh Ridene, Ridha Soua, Erwan Livolant, Ines Khoufi.

deterministic performance, probabilistic performance

Power laws probability distributions that decays has inverse power of the variable for large values of the variable. Power laws are frequent in economic and statistical analysis (see Pareto law). Simple models such as Poisson processes and finite state Markov processes don't generate distributions with power laws.

We develop our performance evaluation tools towards deterministic performance and probabilistic performance. Our tools range from mathematical analysis to simulation and real life experiment of telecommunication algorithms.

One cannot design good algorithms without good evaluation models. Hipercom project team has an historically strong experience in performance evaluation of telecommunication systems, notably when they have multiple access media. We consider two main methodologies:

- Deterministic performance analysis,
- Probabilistic performance analysis

In the deterministic analysis, the evaluation consists to identify and quantify the worst case scenario for an algorithm in a given context. For example to evaluate an end-to-end delay. Mathematically it consists into handling a (max,+) algebra. Since such algebra is not commutative, the complexity of the evaluation of an end-to-end delay frequently grows exponentially with the number of constraints. Therefore the main issue in the deterministic evaluation of performance is to find bounds easier to compute in order to have practical results in realistic situations.

In the probabilistic analysis of performance, one evaluate the behavior of an algorithm under a set of parameters that follows a stochastic model. For example traffic may be randomly generated, nodes may move randomly on a map. The pionneer works in this area come from Knuth (1973) who has systemized this branch. In the domain of telecommunication, the domain has started a significant rise with the appearance of the problematic of collision resolution in a multiple access medium. With the rise of wireless communication, new interesting problems have been investigated.
The analysis of algorithm can rely on analytical methodology which provides the better insight but is practical in very simplistic models. Simulation tools can be used to refine results in more complicated models. At the end of the line, we proceed with real life experiments. To simplify, experiments check the algorithms with 10 nodes in maximum, simulations with 100 nodes maximum, analytical tools with more 1,000 nodes, so that the full range of applicability of the algorithms is investigated.

3.3. Traffic and network architecture modeling

Participants: Cédric Adjih, Aline Carneiro Viana, Emmanuel Baccelli.

traffic source models, network topologies, mobility models, dynamic nodes
Abstract. Network models are important. We consider four model problems: topology, mobility, dynamics and traffic models.

One needs good and realistic models of communication scenarios in order to provide pertinent performance evaluation of protocols. The models must assess the following key points:

- The architecture and topology: the way the nodes are structured within the network
- The mobility: the way the nodes move
- The dynamics: the way the nodes change status
- The traffic: the way the nodes communicate

For the architecture there are several scales. At the internet scale it is important to identify the patterns which dictate the node arrangement. For example the internet topology involves many power law distribution in node degree, link capacities, round trip delays. These parameters have a strong impact in the performance of the global network. At a smaller scale there is also the question how the nodes are connected in a wireless network. There is a significant difference between indoor and outdoor networks. The two kinds of networks differ on wave propagation. In indoor networks, the obstacles such as walls, furniture, etc, are the main source of signal attenuations. In outdoor networks the main source of signal attenuation is the distance to the emitter. This lead to very different models which vary between the random graph model for indoor networks to the unit graph model for outdoor networks.
The mobility model is very important for wireless network. The way nodes move may impact the performance of the network. For example it determines when the network splits in distinct connected components or when these components merge. With random graph models, the mobility model can be limited to the definition of a link status holding time. With unit disk model the mobility model will be defined according to random speed and direction during random times or random distances. There are some minor complications on the border of the map.
The node dynamic addresses the elements that change inside the node. For example its autonomy, its bandwidth requirement, the status of server, client, etc. Pair to pair networks involve a large class of users who frequently change status. In a mobile ad hoc network, nodes may change status just by entering a coverage area, or because some other nodes leaves the coverage area.

The traffic model is very most important. There are plenty literature about trafic models which arose when Poisson models was shown not to be accurate for real traffics, on web or on local area networks. Natural traffic shows long range dependences that don't exist in Poisson traffic. There are still strong issues about the origin of this long range dependences which are debated, however they have a great impact on network performance since congestions are more frequent. The origin are either from the distribution of file sizes exchanged over the net, or from the protocols used to exchange them. One way to model the various size is to consider on/off sources. Every time a node is on it transfers a file of various size. The TCP protocol has also an impact since it keeps a memory on the network traffic. One way to describe it is to use an on/off model (a source sending packets in transmission windows) and to look at the superposition of these on/off sources.

3.4. Algorithm design, evaluation and implementation

Participants: Cédric Adjih, Aline Carneiro Viana, Emmanuel Baccelli, Saoucene Mahfoudh Ridene, Pascale Minet, Paul Mühlethaler, Ridha Soua, Erwan Livolant, Ines Khoufi.

Access protocols, routing, scheduling, QoS

Abstract. Algorithms are conceived with focal point on performance. The algorithms we specify in detail range between medium access control to admission control and quality of service management.

The conception of algorithms is an important focus of the project team. We specify algorithms in the perspective of achieving the best performance for communication. We also strive to embed those algorithms in protocols that involve the most legacy from existing technologies (Operating systems, internet, Wifi). Our aim with this respect is to allow code implementations for real life experiment or imbedded simulation with existing network simulators. The algorithm specified by the project ranges from multiple access schemes, wireless ad hoc routing, mobile multicast management, Quality of service and admission controls. In any of these cases the design emphasize the notions of performance, robustness and flexibility. For example, a flooding technique in mobile ad hoc network should be performing such to save bandwidth but should not stick too much close to optimal in order to be more reactive to frequent topology changes. Some telecommunication problems have NP hard optimal solution, and an implementable algorithm should be portable on very low power processing unit (e.g. sensors). Compromise are found are quantified with respect to the optimal solution.

RAP Project-Team

3. Scientific Foundations

3.1. Design and Analysis of Algorithms

Data Structures, Stochastic Algorithms

The general goal of the research in this domain is of designing algorithms to analyze and control the traffic of communication networks. The team is currently involved in the design of algorithms to allocate bandwidth in optical networks and also to allocate resources in content-centric networks. See the corresponding sections below.

The team also pursues analysis of algorithms and data structures in the spirit of the former Algorithms team. The team is especially interested in the ubiquitous divide-and-conquer paradigm and its applications to the design of search trees, and stable collision resolution protocols.

3.2. Scaling of Markov Processes

The growing complexity of communication networks makes it more difficult to apply classical mathematical methods. For a one/two-dimensional Markov process describing the evolution of some network, it is sometimes possible to write down the equilibrium equations and to solve them. The key idea to overcome these difficulties is to consider the system in limit regimes. This list of possible renormalization procedures is, of course, not exhaustive. The advantages of these methods lie in their flexibility to various situations and to the interesting theoretical problems they raised.

A fluid limit scaling is a particularly important means to scale a Markov process. It is related to the first order behavior of the process and, roughly speaking, amounts to a functional law of large numbers for the system considered.

A fluid limit keeps the main characteristics of the initial stochastic process while some second order stochastic fluctuations disappear. In “good” cases, a fluid limit is a deterministic function, obtained as the solution of some ordinary differential equation. As can be expected, the general situation is somewhat more complicated. These ideas of rescaling stochastic processes have emerged recently in the analysis of stochastic networks, to study their ergodicity properties in particular.

3.3. Structure of random networks

This line of research aims at understanding the global structure of stochastic networks (connectivity, magnitude of distances, etc) via models of random graphs. It consists of two complementary foundational and applied aspects of connectivity.

RANDOM GRAPHS, STATISTICAL PHYSICS AND COMBINATORIAL OPTIMIZATION. The connectivity of usual models for networks based on random graphs models (Erdős–Rényi and random geometric graphs) may be tuned by adjusting the average degree. There is a *phase transition* as the average degree approaches one, a *giant* connected component containing a positive proportion of the nodes suddenly appears. The phase of practical interest is the *supercritical* one, when there is at least a giant component, while the theoretical interest lies at the *critical phase*, the break-point just before it appears.

At the critical point there is not yet a macroscopic component and the network consists of a large number of connected component at the mesoscopic scale. From a theoretical point of view, this phase is most interesting since the structure of the clusters there is expected (heuristically) to be *universal*. Understanding this phase and its universality is a great challenge that would impact the knowledge of phase transitions in all high-dimensional models of *statistical physics* and *combinatorial optimization*.

RANDOM GEOMETRIC GRAPHS AND WIRELESS NETWORKS. The level of connection of the network is of course crucial, but the *scalability* imposes that the underlying graph also be *sparse*: trade offs must be made, which required a fine evaluation of the costs/benefits. Various direct and indirect measures of connectivity are crucial to these choices: What is the size of the overwhelming connected component? When does complete connectivity occur? What is the order of magnitude of distances? Are paths to a target easy to find using only local information? Are there simple broadcasting algorithms? Can one put an end to viral infections? How much time for a random crawler to see most of the network?

NAVIGATION AND POINT LOCATION IN RANDOM MESHES. Other applications which are less directly related to networks include the design of improved navigation or point location algorithms in geometric meshes such as the Delaunay triangulation build from random point sets. There the graph model is essentially fixed, but the constraints it imposes raise a number of challenging problems. The aim is to prove performance guarantees for these algorithms which are used in most manipulations of the meshes.

REGAL Project-Team

3. Scientific Foundations

3.1. Research rationale

Peer-to-peer, Cloud computing, distributed system, data consistency, fault tolerance, dynamic adaptation, large-scale environments, replication.

As society relies more and more on computers, responsiveness, correctness and security are increasingly critical. At the same time, systems are growing larger, more parallel, and more unpredictable. Our research agenda is to design Computer Systems that remain correct and efficient despite this increased complexity and in spite of conflicting requirements.[1]

While our work historically focused on distributed systems, we now cover a larger part of the whole Computer Systems spectrum. Our topics now also include Managed Run-time Environments (MREs, a.k.a. language-level virtual machines) and operating system kernels. This holistic approach allows us to address related problems at different levels. It also permits us to efficiently share knowledge and expertise, and is a source of originality.

Computer Systems is a rapidly evolving domain, with strong interactions with industry. Two main evolutions in the Computer Systems area have strongly influenced our research activities:

3.1.1. Modern computer systems are increasingly distributed.

Ensuring the persistence, availability and consistency of data in a distributed setting is a major requirement: the system must remain correct despite slow networks, disconnection, crashes, failures, churn, and attacks. Ease of use, performance and efficiency are equally important for systems to be accepted. These requirements are somewhat conflicting, and there are many algorithmic and engineering trade-offs, which often depend on specific workloads or usage scenarios.

Years of research in distributed systems are now coming to fruition, and are being used by millions of users of web systems, peer-to-peer systems, gaming and social applications, or cloud computing. These new usages bring new challenges of extreme scalability and adaptation to dynamically-changing conditions, where knowledge of system state can only be partial and incomplete. The challenges of distributed computing listed above are subject to new trade-offs.

Innovative environments that motivate our research include peer-to-peer (P2P) and overlay networks, dynamic wireless networks, cloud computing, and manycore machines. The scientific challenges are scalability, fault tolerance, dynamicity and virtualization of physical infrastructure. Algorithms designed for classical distributed systems, such as resource allocation, data storage and placement, and concurrent access to shared data, need to be revisited to work properly under the constraints of these new environments.

Regal focuses in particular on two key challenges in these areas: the adaptation of algorithms to the new dynamics of distributed systems and data management on large configurations.

3.1.2. Multicore architectures are everywhere.

The fine-grained parallelism offered by multicore architectures has the potential to open highly parallel computing to new application areas. To make this a reality, however, many issues, including issues that have previously arisen in distributed systems, need to be addressed. Challenges include obtaining a consistent view of shared resources, such as memory, and optimally distributing computations among heterogeneous architectures, such as CPUs, GPUs, and other specialized processors. As compared to distributed systems, in the case of multicore architectures, these issues arise at a more fine-grained level, leading to the need for different solutions and different cost-benefit trade-offs.

[1] From the web page of ACM Transactions on Computer Systems: *"The term* 'computer systems' *is interpreted broadly and includes systems architectures, operating systems, distributed systems, and computer networks."* See http://tocs.acm.org/.

Recent multicore architectures are highly diverse. Compiling and optimizing programs for such architectures can only be done for a given target. In this setting, MREs are an elegant approach since they permit distributing a unique binary representation of an application, to which architecture-specific optimizations can be applied late on the execution machine. Finally, the concurrency provided by multicore architectures also induces new challenges for software robustness. We consider this problem in the context of systems software, using static analysis of the source code and the technology developed in the Coccinelle tool.

TREC Project-Team

3. Scientific Foundations

3.1. Scientific Foundations

- **Modeling and performance analysis of wireless networks.** Our main focus was on cellular networks, mobile ad hoc networks (MANETs) and their vehicular variants called VANETs.

  Our main advances about wireless networks have been based on the development of analytical tools for their performance analysis and on new results from network information theory.

  Concerning cellular networks, the main questions bear on coverage and capacity in large CDMA networks when taking intercell interferences and power control into account. Our main focus has been on the design of: 1) a strategy for the densification and parameterization of UMTS and future OFDM networks that is optimized for both voice and data traffic; 2) new self organization and self optimization protocols for cellular networks e.g. for power control, sub-carrier selection, load balancing, etc.

  Concerning MANETs, we investigated MAC layer scheduling algorithms, routing algorithms and power control. The MAC protocols we considered are based on Aloha and CSMA as well as their cognitive radio extensions. We investigated opportunistic routing schemes for MANETs and VANETs. The focus was on cross layer optimizations allowing one to maximize the transport capacity of multihop networks.
- **Theory of network dynamics.** TREC is pursuing the analysis of network dynamics by algebraic methods. The mathematical tools are those of discrete event dynamical systems: semi-rings, and in particular network calculus, ergodic theory, perfect simulation, stochastic comparison, inverse problems, large deviations, etc. Network calculus gives results on worst-case performance evaluation; ergodic theory is used to assess the stability of discrete event dynamical systems; inverse problem methods are used to estimate some network parameters from external observations and to design network probing strategies.
- **The development of stochastic geometry and random geometric graphs tools.** Stochastic geometry is a rich branch of applied probability which allows one to quantify random phenomena on the plane or in higher dimension. It is intrinsically related to the theory of point processes and also to random geometric graphs. Our research is centered on the development of a methodology for the analysis, the synthesis, the optimization and the comparison of architectures and protocols to be used in wireless communication networks. The main strength of this method is its capacity for taking into account the specific properties of wireless links, as well as the fundamental question of scalability.
- **Combinatorial optimization and analysis of algorithms.** In this research direction started in 2007, we build upon our expertise on random trees and graphs and our collaboration with D. Aldous in Berkeley. Sparse graph structures have proved useful in a number of applications from information processing tasks to the modeling of social networks. We obtained new results in this research direction: computation of the asymptotic for the rank of the adjacency matrix of random graphs, computation of the matching number and the b-matching number of large graphs. We also applied our result to design bipartite graph structures for efficient balancing of heterogeneous loads and to analyze the flooding time in random graphs.
- **Economics of networks** The premise of this relatively new direction of research, developed jointly with Jean Bolot [SPRINT ATL and then TECHNICOLOR] is that economic incentives drive the development and deployment of technology. Such incentives exist if there is a market where suppliers and buyers can meet. In today's Internet, such a market is missing. We started by looking at the general problem of security on Internet from an economic perspective.

ALPAGE Project-Team

3. Scientific Foundations

3.1. From programming languages to linguistic grammars

Participants: Éric Villemonte de La Clergerie, Benoît Sagot, Pierre Boullier.

Historically, several members of Alpage were originally specialists in the domain of modeling and parsing for programming languages, and are working for more than 15 years on the generalization and extension of the techniques involved to the domain of natural language. The shift from programming language grammars to NLP grammars seriously increases complexity and requires ways to handle the ambiguities inherent in every human language. It is well known that these ambiguities are the sources of many badly handled combinatorial explosions.

Furthermore, while most programming languages are expressed by (subclasses) of well-understood context-free grammars (CFGs), no consensual grammatical formalism has yet been accepted by the whole linguistic community for the description of human languages. On the contrary, new formalisms (or variants of older ones) appear constantly. Many of them may be classified into the three following large families:

Mildly Context-Sensitive (MCS) formalisms They manipulate possibly complex elementary structures with enough restrictions to ensure the possibility of parsing with polynomial time complexities. They include, for instance, Tree Adjoining Grammars (TAGs) and Multi-component TAGs with trees as elementary structures, Linear Indexed Grammars (LIGs). Although they are strictly more powerful than MCS formalisms, Range Concatenation Grammars (RCGs, introduced and used by Alpage members, such as Pierre Boullier and Benoît Sagot [58], [96], [103]) are also parsable in polynomial time.

Unification-based formalisms They combine a context-free backbone with logic arguments as decoration on non-terminals. Most famous representatives are Definite Clause Grammars (DCGs) where PROLOG powerful unification is used to compute and propagate these logic arguments. More recent formalisms, like Lexical Functional Grammars (LFGs) and Head-Driven Phrasal Structure Grammars (HPSGs) rely on more expressive Typed Feature Structures (TFS) or constraints.

Unification-based formalisms with an MCS backbone The two above-mentioned characteristics may be combined, for instance by adding logic arguments or constraints to non-terminals in TAGs.

An efficient way to develop large-coverage hand-crafted symbolic grammars is to use adequate tools and adequate levels of representation, and in particular Meta-Grammars, one of Alpage's areas of expertise [121], [117]. Meta-Grammars allows the linguist to focus on a modular description of the linguistic aspects of a grammar, rather than focusing on the specific aspects of a given grammatical formalism. Translation from MGs to grammatical formalisms such as TAG or LFG may be automatically handled. Graphical environments can be used to design MGs and their modularity provides a promising way for sharing the description of common linguistic phenomena across human languages.

3.2. Statistical Parsing

Contrary to symbolic approaches to parsing, in statistical parsing, the grammar is extracted from a corpus of syntactic trees : a treebank. The main advantage of the statistical approach is to encode within the same framework the parsing and disambiguating tasks. The extracted grammar rules are associated with probabilities that allow to score and rank the output parse trees of an input sentence. This obvious advantage of probabilistic context-free grammars has long been counterbalanced by two main shortcomings that resulted in poor performance for plain PCFG parsers: (i) the generalization encoded in non terminal symbols that stand for syntagmatic phrases is too coarse (so probabilistic independence between rules is too strong an assertion) and (ii) lexical items are underused. In the last decade though, effective solutions to these shortcomings have

been proposed. Symbol annotation, either manual [84] or automatic [91], [92] captures inter-dependence between CFG rules. Lexical information is integrated in frameworks such as head-driven models that allow lexical heads to percolate up the syntagmatic tree [72], or probabilistic models derived from lexicalized Tree Adjoining grammars, such as Stochastic Tree Insertion Grammars [70].

In the same period, totally different parsing architectures have been proposed, to obtain dependency-based syntactic representations. The properties of dependency structures, in which each word is related to exactly one other word, make it possible to define dependency parsing as a sequence of simple actions (such as read buffer and store word on top of a stack, attach read word as dependent of stack top word, attach read word as governor of stack top word ...) [129], [89]. Classifiers can be trained to choose the best action to perform given a partial parsing configuration. In another approach, dependency parsing is cast into the problem of finding the maximum spanning tree within the graph of all possible word-to-word dependencies, and online classification is used to weight the edges [85]. These two kinds of statistical dependency parsing allow to benefit from discriminative learning, and its ability to easily integrate various kinds of features, which is typically needed in a complex task such as parsing.

Statistical parsing is now effective, both for syntagmatic representations and dependency-based syntactic representations. Alpage has obtained state-of-the-art parsing results for French, by adapting various parser learners for French, and works on the current challenges in statistical parsing, namely (1) robustness and portability across domains and (2) the ability to incorporate exogenous data to improve parsing attachment decisions. We review below the approaches that Alpage has tested and adapted, and the techniques that we plan to investigate to answer these challenges.

In order to investigate statistical parsers for French, we have first worked how to use the French Treebank [55] and derive the best input for syntagmatic statistical parsing [74]. Benchmarking several PCFG-based learning frameworks [11] has led to state-of-the-art results for French, the best performance being obtained with the split-merge Berkeley parser (PCFG with latent annotations) [92].
In parallel to the work on dependency based representation, presented in the next paragraph, we also conducted a preliminary set of experiments on richer parsing models based on Stochastic Tree Insertion Grammars as used in [70] and which, besides their inferior performance compared to PCFG-LA based parser, raise promising results with respect to dependencies that can be extracted from derivation trees. One variation we explored, that uses a specific TIG grammar instance, a *vertical* grammar called *spinal* grammars, exhibits interesting properties wrt the grammar size typically extracted from treebanks (a few hundred unlexicalized trees, compared to 14 000 CFG rules). These models are currently being investigated in our team [113].
Pursuing our work on PCFG-LA based parsing, we investigated the automatic conversion of the treebank into dependency syntax representations [65], that are easier to use for various NLP applications such as question-answering or information extraction, and that are a better ground for further semantic analysis. This conversion can be applied on the treebank, before training a dependency-based parser, or on PCFG-LA parsed trees. This gives the possibility to evaluate and compare on the same gold data, both syntagmatic- and dependency-based statistical parsing. This also paved the way for studies on the influence of various types of lexical information. Results are sketched in section 6.4 .

3.3. Dynamic wide coverage lexical resources

Participants: Benoît Sagot, Laurence Danlos, Rosa Stern, Éric Villemonte de La Clergerie.

Grammatical formalisms and associated parsing generators are useful only when used together with linguistic resources (lexicons, grammars) so as to build operational parsers, especially when considering modern lexically oriented grammatical formalisms. Hence, linguistic resources are the topic of the following section.

However, wide coverage linguistic resources are scarce and expensive, because they are difficult to build, especially when hand-crafted. This observation motivates us to investigate methods, along to manual development techniques, to automatically or semi-automatically acquire, supplement and correct linguistic resources.

Linguistic expertise remains a very important asset to benefit efficiently from such techniques, including those described below. Moreover, linguistically oriented environments with adequate collaborative interfaces are needed to facilitate the edition, comparison, validation and maintenance of large scale linguistic resources. Just to give some idea of the complexity, a syntactic lexicon, as described below, should provide rich information for several tens of thousands of lemma and several hundreds of thousands of forms.

Successful experiments have been conduced by Alpage members with different languages for the automatic acquisition of morphological knowledge from raw corpora [102]. At the syntactic level, work has been achieved on automatic acquisition of atomic syntactic information and automatic detection of errors in the lexicon [130],[10]. At the semantic level, automatic wordnet development tools have been described [95], [123], [82], [80]. All such techniques need of course to be followed by manual validation, so as to ensure high-quality results.

For French, these techniques, and others, have lead some Alpage members to develop one of the main syntactic resources for French, the Le*fff* [98],[8], developed within the Alexina framework, as well as a wordnet for French, the WOLF [7], the first freely available resource of the kind.

In the last few years, Alpage members have shown how to benefit from other more linguistically-oriented resources, such as the Lexique-Grammaire and DICOVALENCE , in order to improve the coverage and quality of the Le*fff* and the WOLF. This work is a good example of how Inria and Paris 7 members of Alpage fruitful collaborate: this collaboration between NLP computer scientists and NLP linguists have resulted in significant advances which would have not been possible otherwise.

Moreover, an increasing effort has been made towards multilingual aspects. In particular, Alexina lexicons developed in 2010 or before exist for Slovak [102], Polish [104], English, Spanish [87], [86] and Persian [108], not including freely-available lexicons adapted to the Alexina framework.

3.4. Shallow processing

Participants: Éric Villemonte de La Clergerie, Benoît Sagot, Rosa Stern.

The constitution of resources such as lexica or grammars raises the issues of the evaluation of these resources to assess their quality and coverage. For this reason, Alpage was the leader of the PASSAGE ANR project (ended in June 2010), which is the follow-up of the EASy parsing evaluation campaign held in 2004 and conducted by team LIR at LIMSI.

However, although developing parsing techniques, grammars (symbolic or probabilistic), and lexica constitute obviously the key efforts towards deep large-scale linguistic processing, these components need to be included inside a full and robust processing chain, able to handle any text from any source. The development of such linguistic chains, such as SXPipe , is not a trivial task [6]. Moreover, when used as a preliminary step before parsers, the quality of parsers' results strongly depends on the quality of such chains. In that regard, less-standard pre-processings such as word clustering have led to promising results [112].

In fact, such processing chains are mostly used as such, and not only as pre-processing tools before parsing. They aim at performing the basic tasks that produce immediately usable results for many applications, such as tokenization, sentence segmentation, spelling correction, and, most importantly, named entity detection, disambiguation and resolution.

3.5. Discourse structures

Participants: Laurence Danlos, Charlotte Roze.

Until now, the linguistic modeling and automatic processing of sentences has been the main focus of the community. However, many applications would benefit from more large-scale approaches which go beyond the level of sentences. This is not only the case for automatic translation: information extraction/retrieval, summarizing, and other applications do need to resolve anaphora, which in turn can benefit from the availability of hierarchical discourse structures induced by discourse relations (in particular through the notion of right frontier of discourse structures). Moreover, discourse structures are required to extract sequential

(chronological, logical,...) or hierarchical representations of events. It is also useful for topic extraction, which in turns can help syntactic and semantic disambiguation.

Although supra-sentential problematics received increasing attention in the last years, there is no satisfying solution to these problems. Among them, anaphora resolution and discourse structures have a far-reaching impact and are domains of expertise of Alpage members. But their formal modeling has now reached a maturity which allows to integrate them, in a near future, inside future Alpage tools, including parsing systems inherited from Atoll.

It is well known that a text is not a random sequence of sentences: sentences are linked the ones to the others by "discourse relations", which give to the text a hierarchical structure. Traditionally, it is considered that discourse relations are lexicalized by connectors (adverbial connectors like *ensuite*, conjunctions like *parce que*), or are not lexicalized. This vision is however too simple:

- first, some connectors (in particular conjunctions of subordination) introduce pure modifiers and must not be considered as bearing discourse relations,
- second, other elements than connectors can lexicalize discourse relations, in particular verbs like *précéder / to precede* or *causer / to cause*, which have facts or fact eventualities as arguments [76].

There are three main frameworks used to model discourse structures: RST, SDRT , and, more recently, D-LTAG. Inside Alpage, Laurence Danlos has introduced D-STAG (Discourse Synchronous TAGs, [77],[5]), which subsumes in an elegant way both SDRT and RST, to the extent that SDRT and RST structures can be obtained by two different partial projections of D-STAG structures. As done in D-LTAG, D-STAG extends a lexicalized TAG analysis so as to deal with the level of discourse. D-STAG has been fully formalized, and is hence possible to implement (thanks to Synchronous TAG, or even TAG parsers), provided one develops linguistic descriptions in this formalism.

AXIS Project-Team (section vide)

IMARA Project-Team

3. Scientific Foundations

3.1. Vehicle guidance and autonomous navigation

Participants: Fawzi Nashashibi, Evangeline Pollard, Benjamin Lefaudeux, Hao Li, Paulo Lopes Resende, Guillaume Tréhard, Pierre Merdrignac, Zayed Alsayed.

There are three basic ways to improve the safety of road vehicles and these ways are all of interest to the project-team. The first way is to assist the driver by giving him better information and warning. The second way is to take over the control of the vehicle in case of mistakes such as inattention or wrong command. The third way is to completely remove the driver from the control loop.

All three approaches rely on information processing. Only the last two involve the control of the vehicle with actions on the actuators, which are the engine power, the brakes and the steering. The research proposed by the project-team is focused on the following elements:

- perception of the environment,
- planning of the actions,
- real-time control.

3.1.1. Perception of the road environment

Either for driver assistance or for fully automated guided vehicles purposes, the first step of any robotic system is to perceive the environment in order to assess the situation around itself. Proprioceptive sensors (accelerometer, gyrometer,...) provide information about the vehicle by itself such as its velocity or lateral acceleration. On the other hand, exteroceptive sensors, such as video camera, laser or GPS devices, provide information about the environment surrounding the vehicle or its localization. Obviously, fusion of data with various other sensors is also a focus of the research. The following topics are already validated or under development in our team:

- relative ego-localization with respect to the infrastructure, i.e. lateral positioning on the road can be obtained by mean of vision (lane markings) and the fusion with other devices (e.g. GPS);
- global ego-localization by considering GPS measurement and proprioceptive information, even in case of GPS outage;
- road detection by using lane marking detection and navigable free space;
- detection and localization of the surrounding obstacles (vehicles, pedestrians, animals, objects on roads, etc.) and determination of their behavior can be obtained by the fusion of vision, laser or radar based data processing;
- simultaneous localization and mapping as well as mobile object tracking using laser-based and stereovision-based (SLAMMOT) algorithms.

This year was the opportunity to focus on two particular topics: SLAMMOT-based techniques and cooperative perception.

3.1.2. 3D environment mapping

Participants: Fawzi Nashashibi, Hao Li, Benjamin Lefaudeux, Paulo Lopes Resende.

In the past few years, we've been focusing on the Disparity map estimation as a mean to obtain dense 3D mapping of the environment. Moreover, many autonomous vehicle navigation systems have adopted stereo vision techniques to construct disparity maps as a basic obstacle detection and avoidance mechanism. Two different approaches where investigated: the Fly algorithm, and the stereo vision for 3D representation.

The Fly Algorithm is an evolutionary optimization applied to stereovision and mobile robotics. Its advantage relies on its precision and its acceptable costs (computation time and resources). In the other approach, originality relies in computing the disparity field by directly formulating the problem as a constrained optimization problem in which a convex objective function is minimized under convex constraints. These constraints arise from prior knowledge and the observed data. The minimization process is carried out over the feasibility set and with a suitable regularization constraint: the Total Variation information, which avoids oscillations while preserving field discontinuities around object edges. Although successfully applied to real-time pedestrian detection using a vehicle mounted stereohead (see LOVe project), this technique could not be used for other robotics applications such as scene modeling, visual SLAM, etc. The need is for a dense 3D representation of the environment obtained with an appropriate precision and acceptable costs (computation time and resources).

Stereo vision is a reliable technique for obtaining a 3D scene representation through a pair of left and right images and it is effective for various tasks in road environments. The most important problem in stereo image processing is to find corresponding pixels from both images, leading to the so-called disparity estimation. Many autonomous vehicle navigation systems have adopted stereo vision techniques to construct disparity maps as a basic obstacle detection and avoidance mechanism. We also worked in the past on an original approach for computing the disparity field by directly formulating the problem as a constrained optimization problem in which a convex objective function is minimized under convex constraints. These constraints arise from prior knowledge and the observed data. The minimization process is carried out over the feasibility set, which corresponds to the intersection of the constraint sets. The construction of convex property sets is based on the various properties of the field to be estimated. In most stereo vision applications, the disparity map should be smooth in homogeneous areas while keeping sharp edges. This can be achieved with the help of a suitable regularization constraint. We propose to use the Total Variation information as a regularization constraint, which avoids oscillations while preserving field discontinuities around object edges.

The algorithm we developed to solve the estimation disparity problem has a block-iterative structure. This allows a wide range of constraints to be easily incorporated, possibly taking advantage of parallel computing architectures. This efficient algorithm allowed us to combine the Total Variation constraint with additional convex constraints so as to smooth homogeneous regions while preserving discontinuities.

Presently, we are currently working on an original stereo-vision based SLAM technique, aimed at reconstructing current surroundings through on-the-fly real time localization of tens of thousands of interest points. This development should also allow detection and tracking of moving objects [1], and is built on linear algebra (through Inria's Eigen library), RANSAC and multi-target tracking techniques, to quote a few.

This technique complements another laser based SLAMMOT technique developed since few years and extensively validated in large scale demonstrations for indoor and outdoor robotics applications. This technique has proved its efficiency in terms of cost, accuracy and reliability.

3.1.3. Cooperative Multi-sensor data fusion

Participants: Fawzi Nashashibi, Hao Li, Evangeline Pollard, Benjamin Lefaudeux, Pierre Merdrignac.

Since data are noisy, inaccurate and can also be unreliable or unsynchronized, the use of data fusion techniques is required in order to provide the most accurate situation assessment as possible to perform the perception task. IMARA team worked a lot on this problem in the past, but is now focusing on collaborative perception approach. Indeed, the use of vehicle-to-vehicle or vehicle-to-infrastructure communications allows an improved on-board reasoning since the decision is made based on an extended perception.

As a direct consequence of the electronics broadly used for vehicular applications, communication technologies are now being adopted as well. In order to limit injuries and to share safety information, research in driving assistance system is now orientating toward the cooperative domain. Advanced Driver Assistance System (ADAS) and Cybercars applications are moving towards vehicle-infrastructure cooperation. In such scenario, information from vehicle based sensors, roadside based sensors and a priori knowledge is generally combined

[1] http://www.youtube.com/watch?v=obH9Z2uOMBI

thanks to wireless communications to build a probabilistic spatio-temporal model of the environment. Depending on the accuracy of such model, very useful applications from driver warning to fully autonomous driving can be performed.

The Collaborative Perception Framework (CPF) is a combined hardware/software approach that permits to see remote information as its own information. Using this approach, a communicant entity can see another remote entity software objects as if it was local, and a sensor object, can see sensor data of others entities as its own sensor data. Last year's developments permitted the development of the basic hardware pieces that ensures the well functioning of the embedded architecture including perception sensors, communication devices and processing tools. The final architecture was relying on the *SensorHub* presented in year 2010 report and demonstrated several times in year 2011 (ITS World Congress, workshop "The automation for urban transport" in La Rochelle...)

Finally, since vehicle localization (ground vehicles) is an important task for intelligent vehicle systems, vehicle cooperation may bring benefits for this task. A new cooperative multi-vehicle localization method using split covariance intersection filter was developed during the year 2012, as well as a cooperative GPS data sharing method.

In the first method, each vehicle estimates its own position using a SLAM approach. In parallel, it estimates a decomposed group state, which is shared with neighboring vehicles; the estimate of the decomposed group state is updated with both the sensor data of the ego-vehicle and the estimates sent from other vehicles; the covariance intersection filter which yields consistent estimates even facing unknown degree of inter-estimate correlation has been used for data fusion.

In the second GPS data sharing method, a new collaborative localization method is proposed. On the assumption that the distance between two communicative vehicles can be calculated with a good precision, cooperative vehicle are considered as additional satellites into the user position calculation by using iterative methods. In order to limit divergence, some filtering process is proposed: Interacting Multiple Model (IMM) is used to guarantee a greater robustness in the user position estimation.

Both methods should be experimentally tested on IMARA veicles in 2013.

3.1.4. Planning and executing vehicle actions

Participants: Plamen Petrov, Joshué Pérez Rastelli, Fawzi Nashashibi, Philippe Morignot, Paulo Lopes Resende, Mohamed Marouf.

From the understanding of the environment thanks to augmented perception, we have either to warn the driver, to help him in the control of his vehicle, or to take control in case of a driverless vehicle. In simple situations, the planning might also be quite simple, but in the most complex situations we want to explore, the planning must involve complex algorithms dealing with the trajectories of the vehicle and its surroundings (which might involve other vehicles and/or fixed or moving obstacles). In the case of fully automated vehicles, the perception will involve some map building of the environment and obstacles, and the planning will involve partial planning with periodical recomputation to reach the long term goal. In this case, with vehicle to vehicle communications, what we want to explore is the possibility to establish a negotiation protocol in order to coordinate nearby vehicles (what humans usually do by using driving rules, common sense and/or non verbal communication). Until now, we've been focusing on the generation of geometric trajectories as a result of a manoeuvre selection process using grid-based rating technique or fuzzy technique. For high speed vehicles, Partial Motion Planning techniques we tested revealed their limitation because of the computational cost. The use of quintic polynomials we designed allowed us to elaborate trajectories with different dynamics adapted to the driver profile. These trajectories have been implemented and validated in DLR's JointSystem demonstrator used in the European project HAVEit as well as in IMARA's electrical vehicle prototype used in the French project ABV. HAVEit was also the opportunity for IMARA to take in charge the implementation of the Co-Pilot system which processes perception data in order to elaborate the high level command for the actuators. These trajectories were also validated on IMARA's cybercars. However, for the low speed cybercars that have pre-defined itineraries and basic manoeuvres it was necessary to develop a more adapted planning and control system. Therefore, we've developed a nonlinear adaptive control for automated overtaking maneuver using

quadratic polynomials and Lyapunov function candidate and taking into account the vehicles kinematics. For the global mobility systems we are developing, controlling the vehicles include also advanced platooning, automated parking, automated docking, etc. Foreach functionality a dedicated control algorithm was designed (see publication of previous years). Today, IMARA is also investigating the opportunity of fuzzy-based control for specific manoeuvres. First results have been recently obtained for reference trajectory following in roundabouts and normal straight roads.

3.2. V2V and V2I Communications for ITS

Participants: Thierry Ernst, Oyunchimeg Shagdar, Gérard Le Lann, Manabu Tsukada, Thouraya Toukabri, Satoru Noguchi, Ines Ben Jemaa, Mohammad Abu Alhoul, Fawzi Nashashibi, Arnaud de la Fortelle.

Wireless communications is expected to play an important role for road safety, road efficiency, and comfort of road users. Road safety applications often require highly responsive and reliable information exchange between neighboring vehicles in any road density condition. Because the performance of the existing radio communications technology largely degrades with the increase of the node density, the challenge of designing wireless communications for safety applications is enabling reliable communications in highly dense scenarios. Targeting this issue, IMARA has been working on medium access control design and visible light communications especially for highly dense scenarios. The works have been carried out considering vehicles' behavior such as vehicles' merging and platooning.

Unlike many of the road safety applications, the applications regarding road efficiency and comfort of road users, on the other hand, often require connectivity to the Internet. Based on our expertise in both Internet-based communications in the mobility context and in ITS, we are now investigating the use of IPv6 (Internet Protocol version 6 which is going to replace the current version, IPv4, in a few years from now) for vehicular communications, in a combined architecture allowing both V2V and V2I. In the context of IPv6, we have been tackling research issues of combinations of MANET and NEMO and Multihoming in Nested Mobile Networks with Route Optimization.

The wireless channel and topology dynamics are the characteristics that require great research challenge in understanding the dynamics and designing efficient communications mechanisms. Targeting this issue we have been working on channel modeling for both radio and visible light communications, and design of communications mechanisms especially for security, service discovery, multicast and geocast message delivery, and access point selection.

Below follows a more detailed description of the related research issues.

3.2.1. Multihoming in nested mobile networks with route optimization

Participants: Manabu Tsukada, Thierry Ernst.

Network mobility has the particularity of allowing recursive mobility, i.e. where a mobile node is attached to another mobile node (e.g. a PDA is attached to the in-vehicle IP network). This is referred to as nested mobility and brings a number of research issues in terms of routing efficiency. Another issue under such mobility configurations is the availability of multiple paths to the Internet (still in the same example, the PDA has a 3G interface and the in-vehicle network has some dedicated access to the Internet) and its appropriate selection.

3.2.2. Service discovery

Participants: Satoru Noguchi, Thierry Ernst.

Vehicles in a close vicinity need to discover what information can be made available to other vehicles (e.g. road traffic conditions, safety notification for collision avoidance). We are investigating both push and pull approaches and the ability of these mechanisms to scale to a large number of vehicles and services on offer.

3.2.3. Geographic multicast addressing and routing

Participants: Ines Ben Jemaa, Oyunchimeg Shagdar, Thierry Ernst, Arnaud de La Fortelle, Fawzi Nashashibi.

Many ITS applications such as fleet management require multicast data delivery. Existing works on this subject tackle mainly the problems of IP multicasting inside the Internet or geocasting in the VANETs. To enable Internet-based multicast services for VANETs, we introduced a framework that: i) to ensure vehicular multicast group reachability through the infrastructure network, defines a distributed and efficient geographic multicast auto-addressing mechanism and ii) to allow simple and efficient data delivery introduces a simplified approach that locally manages the group membership and distributes the packets among them.

3.2.4. Platooning control using visible light communications

Participants: Mohammad Abu Alhoul, Mohamed Marouf, Oyunchimeg Shagdar, Fawzi Nashashibi.

The main purpose of our research is to propose and test new successful supportive communication technology, which can provide stable and reliable communication between vehicles, especially for the platooning scenario. Although that VLC technology has a short history in comparing with other communication technologies, the infrastructure availability and the presence of the congestion in wireless communication channels are proposing VLC technology as reliable and supportive technology which can takeoff some loads of the wireless radio communication. First objective of this work is develop analytical model of VLC to understand its characteristics and limitation. The second objective of this work is to design vehicle platooning control using VLC. In platooning control, a corporation between control and communication is strongly required in order guarantee the platoon's stability (e.g. string stability problem). For this purpose we work on VLC model platooning scenario, to permit each vehicle the trajectory tracking of the vehicle ahead, altogether with a prescribed inter-vehicle distance and considering all the VLC channel model limitations. The integrated channel model to the main Simulink platooning model will be responsible for deciding the availability of the Line-of-Sight for different trajectory's curvatures, which mean the capability of using light communication between each two vehicles in the platooning queue, at the same time the model will calculate all the required parameters acquired from each vehicle controller.

3.2.5. Access point selection

Participant: Oyunchimeg Shagdar.

While 5.9 GHz radio frequency band is dedicated to ITS applications, there is not much known how the channel and network behave in mobile scenarios. In this work we theoretically and experimentally study the radio channel characteristics in vehicular networks, especially the radio quality and bandwidth availability. Based on our study we develop access point selection method to achieve high speed V2I communications.

3.3. Automated driving, intelligent vehicular networks, and safety

Participant: Gérard Le Lann.

Intelligent vehicular networks (IVNs) are one constituent of ITS. IVNs encompass "clusters", platoons and vehicular ad-hoc networks comprising automated and cooperative vehicles. A basic principle that underlies our work is minimal reliance on road-side infrastructures for solving those open problems arising with IVNs. For example, V2V communications only are considered. Trivially, if one can solve a problem P considering V2V communications only, then P is solved with the help of V2I communications, whereas the converse is not true. Moreover, safety in the course of risk-prone maneuvers is our central concern. Since safety-critical scenarios may develop anytime anywhere, it is impossible to assume that there is always a road-side unit in the vicinity of those vehicles involved in a hazardous situation.

3.3.1. Cohorts and groups – Novel constructs for safe IVNs

The automated driving function rests on two radically different sets of solutions, one set encompassing signal processing and robotics (SPR), the other one encompassing vehicular communications and networking (VCN). In addition to being used for backing a failing SPR solution, VCN solutions have been originally proposed for "augmenting" the capabilities offered by SPR solutions, which are line-of-sight technologies, i.e. limited by obstacles. Since V2V omnidirectional radio communications that are being standardized (IEEE 802.11p / WAVE) have ranges in the order of 250 m, it is interesting to prefix risk-prone maneuvers with the exchange of SC messages. Roles being assigned prior to initiating physical maneuvers, the SPR solutions are invoked under favorable conditions, safer than when vehicles have not agreed on "what to do" ahead of time.

VCN solutions shall belong to two categories: V2V omnidirectional (360°) communications and unidirectional communications, implemented out of very-short range antennas of very small beamwidth. This has led to the concept of neighbor-to-neighbor (N2N) communications, whereby vehicles following each other on a given lane can exchange periodic beacons and event-driven messages.

Vehicle motions on roads and highways obey two different regimes. First, stationary regimes, where inter-vehicular spacing, acceleration and deceleration rates (among other parameters), match specified bounds. This, combined with N2N communications, has led to the concept of cohorts, where safety is not at stake provided that no violation of bounds occurs. Second, transitory regimes, where some of these bounds are violated (e.g., sudden braking – the "brick wall" paradigm), or where vehicles undertake risk-prone maneuvers such as lane changes, resulting into SC scenarios. Reasoning about SC scenarios has led to the concept of groups. Cohorts and groups have been introduced in [7] and [31].

3.3.2. Cohorts, N2N communications, and safety in the presence of telemetry failures

In [7] and [31], we show how periodic N2N beaconing serves to withstand failures of directional telemetry devices. Worst-case bounds on safe inter-vehicular spacing are established analytically (simulations cannot be used for establishing worst-case bounds). A result of practical interest is the ability to answer the following question: "vehicles move at high speed in a cohort formation; if in a platoon formation, spacing would be in the order of 3 m; what is the additional safe spacing in a cohort?" With a N2N beaconing period in the range of 100-200 ms, the additional spacing is much less than 1 m. Failure of a N2N communication link translates into a cohort split, one of the vehicles impaired becoming the tail of a cohort, and its (impaired) follower becoming the head of a newly formed cohort. The number of vehicles in a cohort has an upper bound, and the inter-cohort spacing has a lower bound.

3.3.3. Groups, cohorts, and fast reliable V2V Xcasting in the presence of message losses

Demonstrating safety involves establishing strict timeliness ("real time") properties under worst-case conditions (traffic density, failure rates, radio interference ranges). As regards V2V message passing, this requirement translates into two major problems:

- TBD: time-bounded delivery of V2V messages exchanged among vehicles that undertake SC maneuvers, despite high message loss ratios.
- TBA: time-bounded access to a radio channel in open ad hoc, highly mobile, networks of vehicles, some vehicles undertaking SC maneuvers, despite high contention.

Groups and cohorts have proved to be essential constructs for devising a solution for problem TBD. Vehicles involved in a SC scenario form a group where a 3-way handshake is unfolded so as to reach an agreement regarding roles and adjusted motions. A 3-way handshake consists in 3 rounds of V2V Xcasting of SC messages, round 1 being a Geocast, round 2 being a Convergecast, and round 3 being a Multicast. Worst-case time bound for completing a 3-way handshake successfully is in the order of 200 ms, under worst-case conditions. It is well known that message losses are the dominant cause of failures in mobile wireless networks, which raises the following problem with the Xcasting of SC messages. If acknowledgments are not used, it is impossible to predict probabilities for successful deliveries, which is antagonistic with demonstrating safety. Asking for acknowledgments is a non solution. Firstly, by definition, vehicles that are to be reached by a Geocast are unknown to a sender. How can a sender know which acknowledgments to wait for? Secondly, repeating a SC message that has been lost on a radio channel does not necessarily increase chances of successful delivery. Indeed, radio interferences (causing the first transmission loss) may well last longer than 200 ms (or seconds). To be realistic, one is led to consider a novel and extremely powerful (adversary) failure model (denoted Ω), namely the restricted unbounded omission model, whereby messages meant to circulate on f out of n radio links are "erased" by the adversary (the same f links), ad infinitum. Moreover, we have assumed message loss ratios f/n as high as 2/3. This is the setting we have considered in [49], where we present a solution for the fast (less than 200 ms) reliable (in the presence of Ω) multipoint communications problem TBD. The solution consists in a suite of Xcast protocols (the Zebra suite) and proxy sets built out of cohorts. Analytical expressions are given for the worst-case time bounds for each of the Zebra protocols.

Surprisingly, while not being originally devised to that end, it turns out that cohorts and groups are essential cornerstones for solving open problem TBA.

3.4. Managing the system (via probabilistic modeling)

Participants: Guy Fayolle, Cyril Furtlehner, Yufei Han, Arnaud de La Fortelle, Jean-Marc Lasgouttes, Victorin Martin.

The research on the management of the transportation system is a natural continuation of the research of the Preval team, which joined IMARA in 2007. For many years, the members of this team (and of its ancestor Meval) have been working on understanding random systems of various origins, mainly through the definition and solution of mathematical models. The traffic modeling field is very fertile in difficult problems, and it has been part of the activities of the members of Preval since the times of the Praxitèle project.

Following this tradition, the roadmap of the group is to pursue basic research on probabilistic modeling with a clear slant on applications related to LaRA activities. A particular effort is made to publicize our results among the traffic analysis community, and to implement our algorithms whenever it makes sense to use them in traffic management. Of course, as aforementioned, these activities in no way preclude the continuation of the methodological work achieved in the group for many years in various fields: random walks in Z_+^n ([1], [2], [5]), large deviations, birth and death processes on trees, particle systems. The reader is therefore encouraged to read the recent activity reports for the Preval team for more details.

In practice, the group explores the links between large random systems and statistical physics, since this approach proves very powerful, both for macroscopic (fleet management [4]) and microscopic (car-level description of traffic, formation of jams) analysis. The general setting is mathematical modeling of large systems (mostly stochastic), without any a priori restriction: networks [3], random graphs or even objects coming from biology. When the size or the volume of those structures grows (this corresponds to the so-called thermodynamical limit), one aims at establishing a classification based on criteria of a twofold nature: quantitative (performance, throughput, etc) and qualitative (stability, asymptotic behavior, phase transition, complexity).

3.4.1. Exclusion processes

One of the simplest basic (but non trivial) probabilistic models for road traffic is the exclusion process. It lends itself to a number of extensions allowing to tackle some particular features of traffic flows: variable speed of particles, synchronized move of consecutive particles (platooning), use of geometries more complex than plain 1D (cross roads or even fully connected networks), formation and stability of vehicle clusters (vehicles that are close enough to establish an ad-hoc communication system), two-lane roads with overtaking.

Most of these generalizations lead to models that are obviously difficult to solve and require upstream theoretical studies. Some of them models have already been investigated by members of the group, and they are part of wide ongoing research.

3.4.2. Message passing algorithms

Large random systems are a natural part of macroscopic studies of traffic, where several models from statistical physics can be fruitfully employed. One example is fleet management, where one main issue is to find optimal ways of reallocating unused vehicles: it has been shown that Coulombian potentials might be an efficient tool to drive the flow of vehicles. Another case deals with the prediction of traffic conditions, when the data comes from probe vehicles instead of static sensors. Using the Ising model, together with the Belief Propagation algorithm very popular in the computer science community, we have been able to show how real-time data can be used for traffic prediction and reconstruction (in the space-time domain).

This new use of BP algorithm raises some theoretical questions about the properties of the Bethe approximation of Ising models:

- determine the effect of the various variants of BP (in terms of normalization or changes to the Bethe free energy) on the fixed points and their stability;

- find the best way to inject real-valued data in an Ising model with binary variables;
- build macroscopic variables that measure the overall state of the underlying graph, in order to improve the local propagation of information;
- make the underlying model as sparse as possible, in order to improve BP convergence and quality.

IMEDIA2 Team

3. Scientific Foundations

3.1. Introduction

We group the existing problems in the domain of content-based image indexing and retrieval in the following themes: image indexing and efficient search in image collections, pattern recognition and personalization. In the following we give a short introduction to each of these themes.

3.2. Modeling, construction and structuring of the feature space

Participants: Vera Bakic, Nozha Boujemaa, Esma Elghoul, Hervé Goëau, Amel Hamzaoui, Sofiene Mouine, Olfa Mzoughi, Saloua Ouertani-Litayem, Mohamed Riadh Trad, Anne Verroust-Blondet, Itheri Yahiaoui, Zahraa Yasseen.

The goal of IMEDIA2 team is to provide the user with the ability to do content-based search into image databases in a way that is both intelligent and intuitive to the users. When formulated in concrete terms, this problem gives birth to several mathematical and algorithmic challenges.

To represent the content of an image, we are looking for a representation that is both compact (less data and more semantics), relevant (with respect to the visual content and the users) and fast to compute and compare. The choice of the feature space consists in selecting the significant *features*, the *descriptors* for those features and eventually the encoding of those descriptors as image *signatures*.

We deal both with generic databases, in which images are heterogeneous (for instance, search of Internet images), and with specific databases, dedicated to a specific application field. The specific databases are usually provided with a ground-truth and have an homogeneous content (leaf images, for example)

We must not only distinguish generic and specific signatures, but also local and global ones. They correspond respectively to queries concerning parts of pictures or entire pictures. In this case, we can again distinguish approximate and precise queries. In the latter case one has to be provided with various descriptions of parts of images, as well as with means to specify them as regions of interest. In particular, we have to define both global and local similarity measures.

When the computation of signatures is over, the image database is finally encoded as a set of points in a high-dimensional space: the feature space.
A second step in the construction of the index can be valuable when dealing with very high-dimensional feature spaces. It consists in pre-structuring the set of signatures and storing it efficiently, in order to reduce access time for future queries (trade-off between the access time and the cost of storage). In this second step, we have to address problems that have been dealt with for some time in the database community, but arise here in a new context: image databases. Today's scalability issues already put brake on growth of multi-media search engines. The space created by the massive amounts of existing multimedia files greatly exceeds the area searched by today's major engines. Consistent breakthroughs are therefore urgent if we don't want to be lost in data space in ten years. We believe that reducing algorithm complexity remains the main key. Whatever the efficiency of the implementation or the use of powerful hardware and distributed architectures, the ability of an algorithm to scale-up is strongly related to its time and space complexities. Nowadays, efficient multimedia search engines rely on various high level tasks such as content-based search, navigation, knowledge discovery, personalization, collaborative filtering or social tagging. They involve complex algorithms such as similarity search, clustering or machine learning, on heterogeneous data, and with heterogeneous metrics. Some of them still have quadratic and even cubic complexities so that their use in the large scale is not affordable if no fundamental research is performed to reduce their complexities. In this way, efficient and generic high-dimensional similarity search structures are essential for building scalable content-based search systems. Efficient search requires a specific structuring of the feature space (multidimensional indexing, where indexing is understood as data structure) for accelerating the access to collections that are too large for the central memory.

3.3. Pattern recognition and statistical learning

Participants: Nozha Boujemaa, Michel Crucianu, Donald Geman, Wajih Ouertani, Asma Rejeb Sfar.

Statistical learning and classification methods are of central interest for content-based image retrieval. We consider here both supervised and unsupervised methods. Depending on our knowledge of the contents of a database, we may or may not be provided with a set of *labeled training examples*. For the detection of *known* objects, methods based on hierarchies of classifiers have been investigated. In this context, face detection was a main topic, as it can automatically provide a high-level semantic information about video streams. For a collection of pictures whose content is unknown, e.g. in a navigation scenario, we are investigating techniques that adaptively identify homogeneous clusters of images, which represent a challenging problem due to feature space configuration.

Object detection is the most straightforward solution to the challenge of content-based image indexing. Classical approaches (artificial neural networks, support vector machines, etc.) are based on induction, they construct generalization rules from training examples. The generalization error of these techniques can be controlled, given the complexity of the models considered and the size of the training set.

Our research on object detection addresses the design of invariant kernels and algorithmically efficient solutions as well as boosting method for similarity learning. We have developed several algorithms for face detection based on a hierarchical combination of simple two-class classifiers. Such architectures concentrate the computation on ambiguous parts of the scene and achieve error rates as good as those of far more expensive techniques.

Unsupervised clustering techniques automatically define categories and are for us a matter of visual knowledge discovery. We need them in order to:

- Solve the "page zero" problem by generating a visual summary of a database that takes into account all the available signatures together.
- Perform image segmentation by clustering local image descriptors.
- Structure and sort out the signature space for either global or local signatures, allowing a hierarchical search that is necessarily more efficient as it only requires to "scan" the representatives of the resulting clusters.

Given the complexity of the feature spaces we are considering, this is a very difficult task. Noise and class overlap challenge the estimation of the parameters for each cluster. The main aspects that define the clustering process and inevitably influence the quality of the result are the clustering criterion, the similarity measure and the data model.

SMIS Project-Team

3. Scientific Foundations

3.1. Embedded Data Management

The challenge tackled is this research action is twofold: (1) to design embedded database techniques matching the hardware constraints of (current and future) smart objects and (2) to set up co-design rules helping hardware manufacturers to calibrate their future platforms to match the requirements of data driven applications. While a large body of work has been conducted on data management techniques for high-end servers (storage, indexation and query optimization models minimizing the I/O bottleneck, parallel DBMS, main memory DBMS, etc.), less research efforts have been placed on embedded database techniques. Light versions of popular DBMS have been designed for powerful handheld devices yet DBMS vendors have never addressed the complex problem of embedding database components into chips. Proposals dedicated to databases embedded on chip usually consider small databases, stored in the non-volatile memory of the microcontroller –hundreds of kilobytes– and rely on NOR Flash or EEPROM technologies. Conversely, SMIS is pioneering the combination of microcontrollers and NAND Flash constraints to manage Gigabyte(s) size embedded databases. We present below the positioning of SMIS with respect to international teams conducting research on topics which may be connected to the addressed problem, namely work on electronic stable storage, RAM consumption and specific hardware platforms.

Major database teams are investigating data management issues related to hardware advances (EPFL: A. Ailamaki, CWI: M. Kersten, U. Of Wisconsin: J. M. Patel, Columbia: K. Ross, UCSB: A. El Abbadi, IBM Almaden: C. Mohan, etc.). While there are obvious links with our research on embedded databases, these teams target high-end computers and do not consider highly constrained architectures with non traditional hardware resources balance. At the other extreme, sensors (ultra-light computing devices) are considered by several research teams (e.g., UC Berkeley: D. Culler, ITU: P. Bonnet, Johns Hopkins University: A. Terzis, MIT: S. Madden, etc.). The focus is on the processing of continuous streams of collected data. Although the devices we consider share some hardware constraints with sensors, the objectives of both environments strongly diverge in terms of data cardinality and complexity, query complexity and data confidentiality requirements. Several teams are looking at efficient indexes on flash (HP LABS: G. Graefe, U. Minnesota: B. Debnath, U. Massachusetts: Y. Diao, Microsoft: S. Nath, etc.). Some studies try to minimize the RAM consumption, but the considered RAM/stable storage ratio is quite large compared to the constraints of the embedded context. Finally, a large number of teams have focused on the impact of flash memory on database system design (we presented an exhaustive state of the art in a VLDB tutorial [7]). The work conducted in the SMIS team on bi-modal flash devices takes the opposite direction, proposing to influence the design of flash devices by the expression of database requirements instead of running after the constantly evolving flash device technology.

3.2. Access and Usage Control Models

Access control management has been deeply studied for decades. Different models have been proposed to declare and administer access control policies, like DAC, MAC, RBAC, TMAC, and OrBAC. While access control management is well established, new models are being defined to cope with privacy requirements. Privacy management distinguishes itself from traditional access control is the sense that the data to be protected is personal. Hence, the user's consent must be reflected in the access control policies, as well as the usage of the data, its collection rules and its retention period, which are principles safeguarded by law and must be controlled carefully.

The research community working on privacy models is broad, and involves many teams worldwide including in France ENST-B, LIRIS, Inria LICIT, and LRI, and at the international level IBM Almaden, Purdue Univ., Politecnico di Milano and Univ. of Milano, George Mason Univ., Univ. of Massachusetts, Univ. of Texas and Colorado State Univ. to cite a few. Pioneer attempts towards privacy wary systems include the P3P Platform for Privacy Preservation [39] and Hippocratic databases [30]. In the last years, many other policy languages have been proposed for different application scenarios, including EPAL [44], XACML [41] and WSPL [34]. Hippocratic databases are inspired by the axiom that databases should be responsible for the privacy preservation of the data they manage. The architecture of a Hippocratic database is based on ten guiding principles derived from privacy laws.

The trend worldwide has been to propose enhanced access control policies to capture finer behaviour and bridge the gap with privacy policies. To cite a few, Ardagna *et al.* (Univ. Milano) enables actions to be performed after data collection (like notification or removal), purpose binding features have been studied by Lefevre *et al.* (IBM Almaden), and Ni *et al.* (Purdue Univ.) have proposed obligations and have extended the widely used RBAC model to support privacy policies.

The positioning of the SMIS team within this broad area is rather (1) to focus on intuitive or automatic tools helping the individual to control some facets of her privacy (e.g., data retention, minimal collection) instead of increasing the expressiveness but also the complexity of privacy models and (2) to push concrete models enriched by real-case (e.g., medical) scenarios and by a joint work with researchers in Law.

3.3. Tamper-resistant Data Management

Tamper-resistance refers to the capacity of a system to defeat confidentiality and integrity attacks. This problem is complementary to access control management while being (mostly) orthogonal to the way access control policies are defined. Security surveys regularly point out the vulnerability of database servers against external (i.e., by intruders) and internal (i.e., by employees) attacks. Several attempts have been made in commercial DBMSs to strengthen server-based security, e.g., by separating the duty between DBA and DSA (Data Security Administrator), by encrypting the database footprint and by securing the cryptographic material using Hardware Security Modules (HSM) [36]. To face internal attacks, client-based security approaches have been investigated where the data is stored encrypted on the server and is decrypted only on the client side. Several contributions have been made in this direction, notably by U. of California Irvine (S. Mehrotra, Database Service Provider model), IBM Almaden (R. Agrawal, computation on encrypted data), U. of Milano (E. Damiani, encryption schemes), Purdue U. (E. Bertino, XML secure publication), U. of Washington (D. Suciu, provisional access) to cite a few seminal works. An alternative, recently promoted by Stony Brook Univ. (R. Sion), is to augment the security of the server by associating it with a tamper-resistant hardware module in charge of the security aspects. Contrary to traditional HSM, this module takes part in the query computation and performs all data decryption operations. SMIS investigates another direction based on the use of a tamper-resistant hardware module on the client side. Most of our contributions in this area are based on exploiting the tamper-resistance of secure tokens to build new data protection schemes.

While our work on Privacy-Preserving data Publishing (PPDP) is still related to tamper-resistance, a complementary positioning is required for this specific topic. The primary goal of PPDP is to anonymize/sanitize microdata sets before publishing them to serve statistical analysis purposes. PPDP (and privacy in databases in general) is a hot topic since 2000, when it was introduced by IBM Research (R. Agrawal : IBM Almaden, C.C. Aggarwal: IBM Watson), and many teams, mostly north American universities or research centres, study this topic (e.g., PORTIA DB-Privacy project regrouping universities such as Stanford with H. Garcia-Molina). Much effort has been devoted by the scientific community to the definition of privacy models exhibiting better privacy guarantees or better utility or a balance of both (such as differential privacy studied by C. Dwork : Microsoft Research or D. Kifer : Penn-State Univ and J. Gehrke : Cornell Univ) and thorough surveys exist that provide a large overview of existing PPDP models and mechanisms [40]. These works are however orthogonal to our approach in that they make the hypothesis of a trustworthy central server that can execute the anonymization process. In our work, this is not the case. We consider an architecture composed of a large

population of tamper-resistant devices weakly connected to an untrusted infrastructure and study how to compute PPDP problems in this context. Hence, our work has some connections with the works done on Privacy Preserving Data Collection (R.N.Wright : Stevens Institute of Tech. / Rutgers Univ, NJ, V. Shmatikov : Univ Austin Texas), on Secure Multi-party Computing for Privacy Preserving Data Mining (J. Vaidya : Rutgers Univ, C. Clifton : Purdue Univ) and on distributed PPDP algorithms (D. DeWitt : Univ Wisconsin, K. Lefevre : Univ Michigan, J. Vaidya : Rutgers Univ, C. Clifton : Purdue Univ) while none of them share the same architectural hypothesis as us.

WILLOW Project-Team

3. Scientific Foundations

3.1. 3D object and scene modeling, analysis, and retrieval

This part of our research focuses on geometric models of specific 3D objects at the local (differential) and global levels, physical and statistical models of materials and illumination patterns, and modeling and retrieval of objects and scenes in large image collections. Our past work in these areas includes research aimed at recognizing rigid 3D objects in cluttered photographs taken from arbitrary viewpoints (Rothganger *et al.*, 2006), segmenting video sequences into parts corresponding to rigid scene components before recognizing these in new video clips (Rothganger *et al.*, 2007), retrieval of particular objects and buildings from images and videos (Sivic and Zisserman, 2003) and (Philbin *et al.*, 2007), and a theoretical study of a general formalism for modeling central and non-central cameras using the formalism and terminology of classical projective geometry (Ponce, 2009 and Batog *et al.*, 2010).

We have also developed multi-view stereopsis algorithms that have proven remarkably effective at recovering intricate details and thin features of compact objects and capturing the overall structure of large-scale, cluttered scenes. We have obtained a US patent 8,331,615 [1] for the corresponding software (PMVS, http://grail.cs.washington.edu/software/pmvs/) which is available under a GPL license and used for film production by ILM and Weta as well as by Google in Google Maps. It is also the basic technology used by Iconem, a start-up founded by Y. Ubelmann, a Willow collaborator.

for free for academics, and licensing negotiations with several companies are under way.

Our recent work (Russel *et al.*, 2011) has applied our multi-view-stereo approach to model archaeological sites together with developing representations and efficient retrieval techniques to enable matching historical paintings to 3D models of archaeological sites. This direction is currently being continued in the PhD work of Mathieu Aubry. Our other current work outlined in detail in Section 6.1 is focused on (i) recovering indoor scene geometry from observations of person-object interactions video, (ii) visual place recognition in structured databases, where images are geotagged and organized in a graph, and (iii) developing a discriminative clustering approach able to discover geographically representative image elements from Google Street View imagery using only weak geographic supervision.

3.2. Category-level object and scene recognition

The objective in this core part of our research is to learn and recognize quickly and accurately thousands of visual categories, including materials, objects, scenes, and broad classes of temporal events, such as patterns of human activities in picnics, conversations, etc. The current paradigm in the vision community is to model/learn one object category (read 2D aspect) at a time. If we are to achieve our goal, we have to break away from this paradigm, and develop models that account for the tremendous variability in object and scene appearance due to texture, material, viewpoint, and illumination changes within each object category, as well as the complex and evolving relationships between scene elements during the course of normal human activities. Our current work, outlined in detail in Section 6.2), focuses on the two problems described next.

[1] The patent "Match, Expand, and Filter Technique for Multi-View Stereopsis," was issued December 11, 2012 and assigned patent number 8,331,615.

3.2.1. Learning image and object models.

Learning sparse representations of images has been the topic of much recent research. It has been used for instance for image restoration (e.g., Mairal *et al.*, 2007) and it has been generalized to discriminative image understanding tasks such as texture segmentation, category-level edge selection and image classification (Mairal *et al.*, 2008). We have also developed fast and scalable optimization methods for learning the sparse image representations, and developed a software called SPAMS (SPArse Modelling Software) presented in Section 5.2 . The work of J. Mairal is summarized in his thesis (Mairal, 2010). The most recent work has focused on developing a general formulation for supervised dictionary learning and investigating methods to learn better mid-level features for recognition.

3.2.2. Category-level object/scene recognition and segmentation

Another significant strand of our research has focused on the extremely challenging goals of category-level object/scene recognition and segmentation. Towards these goals, we have developed: (i) strongly-supervised deformable part-based model for object recognition and localization, (ii) a MRF model for segmentation of text in natural scenes, and (iii) algorithms for multi-class cosegmentation using a novel energy-minimization approach based on the developed convex relaxation for weakly supervised classifiers.

3.3. Image restoration, manipulation and enhancement

The goal of this part of our research is to develop models, and methods for image/video restoration, manipulation and enhancement. The ability to “intelligently" manipulate the content of images and video is just as essential as high-level content interpretation in many applications: This ranges from restoring old films or removing unwanted wires and rigs from new ones in post production, to cleaning up a shot of your daughter at her birthday party, which is lovely but noisy and blurry because the lights were out when she blew the candles, or editing out a tourist from your Roman holiday video. Going beyond the modest abilities of current “digital zoom" (bicubic interpolation in general) so you can close in on that birthday cake, “deblock" a football game on TV, or turn your favorite DVD into a blue-ray, is just as important.

In this context, we believe there is a new convergence between computer vision, machine learning, and signal processing. For example: The idea of exploiting self-similarities in image analysis, originally introduced in computer vision for texture synthesis applications (Efros and Leung, 1999), is the basis for non-local means (Buades *et al.*, 2005), one of today’s most successful approaches to image restoration. In turn, by combining a powerful sparse coding approach to non-local means (Dabov *et al.*, 2007) with modern machine learning techniques for dictionary learning (Mairal *et al.*, 2010), we have obtained denoising and demosaicking results that are the state of the art on standard benchmarks (Mairal *et al.*, 2009).

Our current work, outlined in detail in Section 6.3 , has focused on (i) developing a geometrical model for removing image blur due to camera shake, (ii) preparing an online image deblurring demo, and (iii) developing new formulation for image deblurring cast as a multi-label energy minimization problem.

3.4. Human activity capture and classification

From a scientific point of view, visual action understanding is a computer vision problem that until recently has received little attention outside of extremely specific contexts such as surveillance or sports. Many of the current approaches to the visual interpretation of human activities are designed for a limited range of operating conditions, such as static cameras, fixed scenes, or restricted actions. The objective of this part of our project is to attack the much more challenging problem of understanding actions and interactions in unconstrained video depicting everyday human activities such as in sitcoms, feature films, or news segments. The recent emergence of automated annotation tools for this type of video data (Everingham, Sivic, Zisserman, 2006; Laptev, Marszałek, Schmid, Rozenfeld, 2008; Duchenne, Laptev, Sivic, Bach, Ponce, 2009) means that massive amounts of labelled data for training and recognizing action models will at long last be available. Our research agenda in this scientific domain is described below and our recent results are outlined in detail in Section 6.4 .

3.4.1. Weakly-supervised learning and annotation of human actions in video

We aim to leverage the huge amount of video data using readily-available annotations in the form of video scripts. Scripts, however, often provide only imprecise and incomplete information about the video. We address this problem with weakly-supervised learning techniques both at the text and image levels. To this end we recently explored automatic mining of scene and action categories. Within PhD of Piotr Bojanowski we are currently extending this work towards exploiting richer textual descriptions of human actions and using them for learning more powerful contextual models of human actions in video.

3.4.2. Descriptors for video representation

Video representation has a crucial role for recognizing human actions and other components of a visual scene. Our work in this domain aims to develop generic methods for representing video data based on realistic assumptions. We explore the ways of enriching standard bag-of-feature representations with the higher-level information on objects, scenes and primitive human actions pre-learned on related tasks. We also investigate highly-efficient methods for computing video features motivated by the need of processing very large and increasing amounts of video.

3.4.3. Crowd characterization in video

Human crowds are characterized by distinct visual appearance and require appropriate tools for their analysis. In our work we develop generic methods for crowd analysis in video aiming to address multiple tasks such as (i) crowd density estimation and localization, (ii) characterization and recognition of crowd behaviours (e.g a person running against the crowd flow) as well as (iii) detection and tracking of individual people in the crowd. We address the challenge of analyzing crowds under the large variation in crowd density, video resolution and scene structure.

3.4.4. Modeling and recognizing person-object and person-scene interactions.

Actions of people are tightly coupled with their environments and surrounding objects. Moreover, object function can be learned and recognized from observations of person-object interactions in video and still images. Designing and learning models for person-object interactions, however, is a challenging task due to both (i) the huge variability in visual appearance and (ii) the lack of corresponding annotations. We address this problem by developing weakly-supervised techniques enabling learning interaction models from long-term observations of people in natural indoor video scenes such as obtained from time-lapse videos on YouTube. We also explore stereoscopic information in 3D movies to learn better models for people in video including person detection, segmentation, pose estimation, tracking and action recognition.